MSX MAGAZINE

永久保存版

MSXPLAYer 初の公式ガイドブック

キーパーソン語る
西　和彦

懐かしのコミックも復活！
ウーくんのソフト屋さん
MSXPLAYerあらし
のんきな父さん

Windowsに蘇る 名作ゲーム20本収録

ザ・キャッスル／ボコスカウォーズ／テグザー
ファイアーホーク／イースシリーズ体験版など

ASCII

Pocket PC 2002版も付属

10年の沈黙を破り、ついに伝説が復活する...

でんせつ⓪【伝説】歴史上の人物や、具体的な事物・事件などに関する言い伝え。〔広義では、事実よりも真実な話柄ﾜﾍとしてその社会に広まっている事柄を指す。
例、「A教授が厳しいことは、すでに――となっている」「修業にまつわる――的な〔＝ひとつ話として語り伝えられる〕苦労話 ――的〔＝実在した証明は無いが、その性格・活躍が生き生きとして語り伝えられる〕英雄――上の△人物（動物）」

（新明解国語辞典 第五版 三省堂より）

CONTENTS

Cover Photo:Hideki Ichihara

MSXPLAYerで伝説の8ビットパソコンが蘇る

復活！MSX

Windowsパソコンや、Pocket PC 2002上でMSXの懐かしのゲームがプレイできる公式MSXエミュレータMSXPLAYerがついに登場した。しかもWindows版にはMSX-BASICで自由にプログラムが組める「BASIC版」も用意されている。君のパソコン上で、あの伝説の8ビットパソコンMSXを蘇らせよう！

リビングに置いて、みんなでワイワイやれるパソコンなんて、いまどきないよね。でもMSXはそんなパソコンだったんだ。MSXPLAYerがあれば、うちのノートパソコンも、きっとそんな楽しいパソコンに変身するんじゃないかな。

伝説の 8ビットパソコン MSX

1983年、8ビットパソコンの世界統一規格としてMSXが登場した。ソニー、富士通などそれ以前から8ビットパソコンを製造・販売していたメーカーはもちろん、それまでパソコンとは縁の薄かった家電メーカーまでもがMSX規格のパソコンを発売した。海外においても、オランダのフィリップス、韓国の大宇、SAMSUNGなど、名だたるメーカーがMSXパソコンを発売した。

手ごろな価格、オープンアーキテクチャによる開発のしやすさ、ROMカートリッジ採用による遊びやすさなどが人気を呼び、MSXは多くのパソコン少年、パソコン少女を生み出し、そして育てた。

規格自体も時代と共に拡張され、MSXからMSX2、MSX2＋、MSX turboRへと進化を遂げていった。

しかし、1991年、MSX turboR規格のパソコン「FS-A1GT」の発売を最後に、MSXの新製品が登場することはなくなり、やがてパソコン市場から姿を消していったのである。

それから、十余年の歳月が流れた……。

2002年末、伝説となった8ビットパソコンがいよいよ復活する！

MSX PLAYerで蘇るMSX

MSXPLAYerは、MSXパソコンのハードウェアとシステムソフトウェアをすべてソフトウェアで再現するエミュレータだ。しかも、MSXのBIOSを正式に使用許諾された「公式エミュレータ」なのだ。

MSXPLAYerには、現在、Windows版とPocket PC版が用意されている。つまり、あなたのWindowsパソコンやPocket PC上でMSXPLAYerを動かせば、そのまた上で、MSX用のソフトウェアが動いてしまうというわけだ。

「ボコスカウォーズ」や「ザ・キャッスル」など、MSXで人気を博したゲームが、当時のテイストそのままに遊べるようになる。

平日は32ビット2.53GHzでバリバリと仕事をこなし、休日は8ビット3.58MHzでゆっくりとした時間を過ごす。初めてプログラムを組む楽しみを教えてくれたMSX-BASICと、久しぶりに対話してみようか。

まずは、本書の付属CD-ROMに収録された懐かしの8ビットゲームたちを遊んでみてほしい、寝食を忘れて熱中したあの頃に帰れること間違いなし！

さらに、Windows版には、MSX-BASICが起動するBASIC版も用意されている。昔遊んだ懐かしいあのプログラムから、昔挑んだ恥ずかしい自作プログラム（失礼！）まで、Windows上で動かせるというわけ。もちろん、今からBASICでプログラミングにチャレンジすることもできるぞ。

さぁ、MSXPLAYerを使って、新しくて懐かしい、伝説の8ビットパソコンを存分に味わってみよう。

協力：株式会社ソーテック、株式会社東芝、日本ヒューレット・パッカード株式会社（順不同）

復活!MSX

天気のいい日には、ポケットに懐かしのゲームを忍ばせて、お出かけ。お気に入りの場所に着いたら、ボタン1つでタイムスリップ。これって私だけのタイムマシンなのかも。

MSXがこの世になかったら、僕のPC生活はどうなっていただろう？　毎日1000以上のヒット数を誇る自作のホームページも、フルタワーの自作PCも、絶対なかったに違いない。それこそ僕はキーボード恐怖症の、ただのオヤジになっていたかもしれない！MSXは僕のPCライフの原点なのだから！　MSX復活バンザイ！！

声優：古谷 徹

月刊アスキー編集長：遠藤諭

パソコンというのは半分くらいはその話をするためにあるようなものだと思う。みんなで同じものを使っていれば、そこから生み出されるものは、使う人の数の2乗になっていく（メトカルフェの法則？）。MSXは、そんな夢を運んだ。だからいままた語られるべきなのだ。

マガジンハウス：広瀬桂子
（初代「ウーくんのソフト屋さん」担当）

「暗号製造マシン」だの「福笑いで初笑い」だの、実に他愛のないネタばかり、よく毎月思いついたものだと我ながら呆れ。月に何度も「動かない」と半泣きの子供から電話を受け「コンマもちゃんと打ち込んでね」としか言えなかった、プログラムのプの字もわからない担当でした……ウーくん復活おめでとー！

ビッツラボラトリー：宮崎暁

「MSX2テクニカル・ハンドブック」「MSX-DOSスーパーハンドブック」あるいはポケットバンクシリーズを執筆していたころが懐かしい。今回、当時のプログラムが収録されるということで「まさか」という思いです。MSXPLAYerをボクも是非動かしてみたいな。スタッフのみなさん、ごくろうさま。

MSX復活+Mマガ復活
お祝いメッセージ

クリスマスよりも紅白歌合戦よりも、
そして正月よりも素晴らしい日がやってきた。
MSXそしてMマガの復活だ。
バンザ～イ！
そんなめでたい今をたたえて、MSXゆかりの方々からも、
ぞくぞくと"お祝いメッセージ"を頂戴したのだ。
みなさん、どうもありがとうございます。
これをはげみにして今後もガンガン盛り立てるぞ！ おー！

漫画家：すがやみつる

株式会社ぴえろ：田口旬一
（初代「MSXマガジン」編集長）

まさかMSXマガジンが復活するなんて……。思い起こせば若気の至りで創刊してしまったMSXマガジンに人生の中で再会できるとは思ってもいませんでした。実に嬉しい限りです。復活に携わったスタッフのみなさん、本当にありがとうございます＆ご苦労様でした。

漫画家：**桜玉吉**

MSX!!
ああ何て懐しくも
香わしいひびきでしょう!
MSXマガジンが無かったら
今の私、
桜玉吉は
存在しなかったと断言できます!

今、ひきこもりの病人だけどな。

復活おめでとうございます!!
ウーくんも大喜びです!
やあ!

漫画家：**桜沢エリカ**

ボーステック株式会社 代表取締役：八巻龍一

ボーステックの第1作目のソフトは実は『ANTY』というMSXのゲームソフトだったんです。「MSXマガジン」が復活されるということは大変喜ばしいです。当社も多くのMSXソフトを作成しました。「プロジェクトEGG」で公開していますのでよろしくお願いします。

漫画家：渡辺健一

Mマガ復活おめでとうございます。MSX、そしてMマガを初めて買った時のドキドキは今でも覚えています。MSXの思想は今も有効だと思います。がんばっていきましょう。

漫画家：花見沢Q太郎

MSXマガジン復刊おめでとうございます!! からくり道中!! 発売を待ちわびてTAKERU僕の心です。あれから僕はピンクソックスもできるほどに、大人になりましたが（当時もやりましたが）いつも僕の心のスクリーン2にはMSXがいました。今回はおまけにMSX本体（MSXPLAYer）もついてくるそうで私の人生シッチャカメッチャカにされそうです。

マイクロソフト株式会社 ニューメディア&デジタルデバイス本部：御代茂樹

MSXが、WindowsやPocketPC上のソフトウェアとして復活したことをうれしく思います。これにより、WindowsやPocketPCの市場がさらに活性化することを期待しています。

モオソフト：青井大地

MSXマガジン復活&MSXPLAYer正式リリースおめでとうございます。しばらく新作は出していませんが、まだまだ現役MSXクリエイターを自負しております。みんな！ これからも一緒にがんばって行こう！！

ミュージシャン：横川理彦

（よっちゃん、元MSX・FANライター）

あー、MSX復活、おめでとございます。感想は、そうか、みんな大人になったなーということでしょうか。やっぱ、アレやね、好きなことをやるに限るよ。大人だし。

海外からも応援だ！

韓国のプロレスラー：キム・ナムフン

小学生のとき、ソウルでこっそり売ってる「MSXマガジン」を買いに、高速バスに乗って、地下鉄に乗って、3時間くらいかかって行ってました。いまでも、ソニー、松下、そして韓国の大宇のMSXを持ってますよ。MSXとMSXマガジン復活、本当にオメデトウございます。

ゴキゲン8bitゲーム

～青春の懐ゲーが今ここに復活～

世の中、3次元グラフィックスがグリグリ動き、PCM音源によるハイクオリティすぎな音楽と効果音がジャカスカ響き、メモリをドカンと使いまくっているゲームばかりが目に付くが、過去に作られたゲームだって迫力の点ではチョイとかなわないものの、アイデアに優れた愛すべきタイトルがいろいろあったのだ。その中にはいまだに語り継がれる名作も少なくない。そうしたゲームがただ歴史の中に埋もれてしまっていいモノだろうか。

今回、MSXPLAYerという最強のMSXエミュレータを得たMマガとしては、ともかく世間から忘れ去られようとしているMSXゲームを、少しでも復活させたいという思いにかられ、限られた時間の中、ともかく動いてみることにした。しかし、やってみると著作者に連絡がとれなかったり、メーカーが消えていたり、権利関係が不明だったりと、ほとんど探偵になったような、そんな作業の日々の連続。正直言って、10年の月日の掘り起こしがこれほどツラく大変なコトだとは思わなかったぞ（マジだ）。

しかし、とにもかくにもこうして著作者のご理解をいただき、いくつかのタイトルの掲載にこぎつけることができた。むろん、あれがやりたかった、これがやりたかったというご意見もあるだろうが、これが公然たるMSXゲーム復活の第一歩となることを願って、まずは素直に喜んでいただきたい。つーわけでね、レッツ・プレイ・エブリワンっす！

収録ゲーム・リスト

注：PocketPCについては、操作が無理なくできるものだけをチョイスした。それでも機種によってプレイしやすさに違いがあることをご了承いただきたい。
注：体験版は20分の時間制限がある。また、データのロード/セーブができない。

Illustrator：Suguru.T

これらのゲームソフトは付属CD-ROMからインストールして遊ぶことができる。インストールする方法については本誌「スタートMSXPLAYer」(p.52)から解説しているので、そちらをご覧いただきたい。

100の部屋を攻略してお姫様を助けよう

ザ・キャッスル

The Castle

Windows　Pocket PC

1986年 発売：アスキー
作者：吉田功、岩倉啓祐

これは、この世がまだ剣と魔法の力で動かされ、男には愛の証として知恵と勇気とが求められていたころのお話。多くの人から愛されていた「マルガリータ姫」が、グロッケン城に住む「魔王メフィスト」に連れ去られてしまった。彼女を救おうと若者たちが城に入って行ったが、魔王の手下たちにより阻まれるばかり。ある日隣の国の、背が低くて気弱な「ラファエル王子」がやってきた。勇気を出して魔王の城に挑もうというのだ。伝説によると、城内には森の妖精が捕らえられているという。このひ弱な王子は、妖精を味方にしてマルガリータ姫が救出できるだろうか。

扉を開けば未来が開く

城のどこかに閉じ込められているお姫様を、ラファエル王子が助けに行くのがゲームの目的。一見かわいいキャラクタばかりだし、音楽ものどか。しかし、多くの人を唸らせたゲームだけあって、おもしろさと難しさは想像以上だぞ。

まずマップから解説しよう。城は縦10×横10＝計100の部屋で仕切られていて、地下3階、地上7階で構成される。最初は1階の入り口からスタートして、地下そして上階へと進むことになる。各部屋の移動には全6色ある扉を開けなければならない。それには扉と同じ色の鍵が必要だ。開けるには扉の方向に進むだけ。同じ色の鍵を持っていれば自動的に開いてくれる。一度使った鍵はなくなってしまうが、鍵は城内を歩き回っているうちに入手できる。最初のうちは拾いそこねると鍵が足りなくなる可能性があるから、見かけたらなるべく拾うこと。一部の鍵は妖精が持っていたりもするぞ。

ジャンプ技を極めよう

デカ帽子をかぶったラファエル王子はジャンプが得意。なんと、横方向には身長の6倍も飛べるのだ。さらに、縦方向には身長の3倍分もジャンプできる。これで敵を飛び越えたり、高い段に飛び乗ったりと、いろいろなアクションが可能となる。ジャンプの軌跡が、斜め上→水平→斜め下と、山形になっているのはご愛嬌だが、逆にそれを利用したジャンプテクニックがあり、これを利用しないとクリアできないところが何か所もあるのだ。そこで基本的なジャンプ技を示しておいたので参考にしてほしい。それと、王子は金庫、樽、レンガ、壺などの品物を押して移動させることができる。足場を作ったり敵を押し潰したりするのに使えるぞ。その他ザ・キャッスルには、さまざまなルールが存在する。たとえば、壺を移動しても一度別の部屋に行ってくれば元の位置に戻るとか、エレベータの下は通り抜けられる、などだ。これらは経験で理解してほしい。あとは勇気とタイミングと頭脳が勝負！お姫様を助けるまでガンバレ！

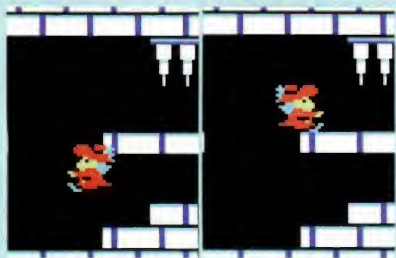
ゲーム解説

ザ・キャッスルの操作

キー	意味
[←][→]キー	左右に動く
[スペース]キー	ジャンプ
[CTRL]キー	速度アップ
[CTRL]＋[GRAPH]キー	速度アップ(大)
[F1]キー	自殺
[F2]キー	ゲームオーバー
[F3]キー	データ読み込み
[F4]キー	データ保存

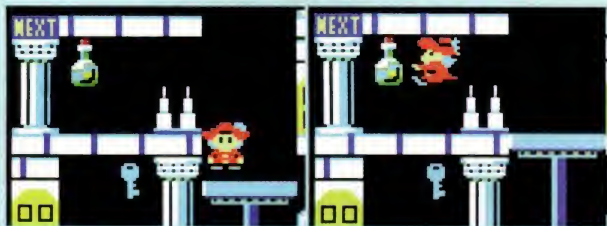
注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する

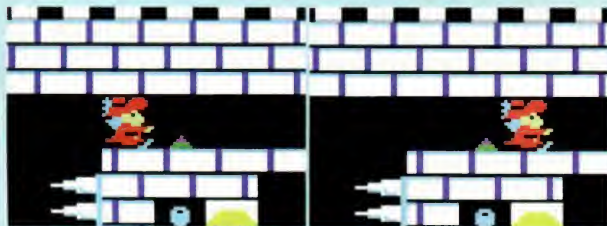
起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

王子の華麗なスーパージャンプ

必殺！三角飛びっ

なんとジャンプ中に空中でクイッと方向を変えることができる。プレイヤーの巧みな指の動きが王子の生死を左右するぞ。

これが直角飛びだ！

真上に飛び上がって横に進む。直角飛びを利用すれば針の上でも安全。

頭スリスリ飛びー！

頭を天井にぶつけて、そのまま跳び続ける。これも大事なワザ！

one point column ワンポイントコラム

途中セーブで保険をかけろ

製品版のザ・キャッスルには途中セーブする機能があった。ところがセーブ先はテープだったので、テープに非対応のMSXPLAYerでは、この機能を利用することはできない。しかし、これでは辛すぎる。そこで本書のMSXPLAYer版ザ・キャッスルはハードディスクに保存できるようにしてあるのだ。使い方は製品版とまったく同じ、[F3]キーでデータ読み込み、[F4]キーでデータ保存だ。難易度の高い部屋の前で使ってほしい。ただし、途中セーブには命を1つ使ってしまうので肝心なときだけ使ってね。

[F4]キーを押すと、このように表示されるので、[Y]キーを押して保存しよう。ただし、一応保存できるのは1か所だけだぞ。

動かせないアイテム

鍵

鍵と同じ色の扉が開けられる。水色や黄色などの淡色の鍵よりも、赤や青などの濃色の鍵ほど重要度が高い。

マップ

取ると画面上部の「NO MAP」という部分に地図が現れ、現在位置が白く表示される。一度訪れた部屋は青くなる。

命の水

取ると命が1つ増える。これが置いてある場所は「そろそろ難しい場面だからセーブ希望」という意味かも？

空気タンク

すぐに溺れる王子が水中を進めるようになる。でも制限時間があるので王子の点滅が早くなってきたら要注意！

チェリー

真っ赤なサクランボ。取るとすべてのドアが開く。キャッスルエクセレントでは取れないところにあるのだ。

指輪、金塊、十字架

得点になるアイテムだ。1万点で命(ライフ)が1つ増えるので地道な得点稼ぎも重要である。でも無理はしないでね。

登場キャラ

ラファエル王子

プレイヤーキャラ。唯一の得意技はジャンプのみだが、妙に滞空時間が妙に長い上、ピロロロローッという音つきだ。

森の妖精

お城の中に捕らえられている妖精。助けるとお姫様を助け出すお手伝いをしてくれるのだ。お手伝いの内容はひみつ。

騎士

ツルハシを振り上げているロボットのように見えるが、実は盾と矛を持った騎士なのだ。あまりたいした敵じゃないぞ。

魔法使い

杖を持った魔法使いだが特に魔法は使ってこない。ただしエレベータに乗ってくるので、うっかりすると相乗りだ。

食獣植物

小さな芽があっという間に成長する恐ろしい植物。しばらくすると芽に戻る。芽のときにジャンプすると飛び越せる。

司教

小脇に聖書を抱えた司教だが、どうも悪徳司教のようだ。ジャンプして越えるか、金庫、樽、レンガ、壺でつぶそう。

炎

グロッケン城で一番手強い敵。空を飛んでラファエル王子に襲いかかってくる。とにかくやり過ごして逃げること。

戦士

ニャンコ大王やゾウだとの声もあるが、本当はバイキングの戦士なのだ。高い所にいても近づくと落ちてくる。

お姫様

城のどこかに幽閉されているお姫様。ラファエル王子の一途な気持ちは果たして姫に届くのだろうか。とどけー。

※主な設備と動くアイテムについては、次の「キャッスルエクセレント」で解説しているのだ。

作者から一言

本作品はアスキーソフトウェアコンテストでグランプリを頂いたゲームでした。当時の8ビット機種の多くに移植されましたが、MSX版への移植は「キャッスルエクセレント」を作った岩倉さんにお願いしたもので、できがもっともよかったと記憶しています。「ザ・キャッスル」のBGMは、今でも携帯電話の着メロとして設定してます。そういえば、キャッスルエクセレントは未だにクリアできていません。10年ぶりにMSXPLAYerで再度トライしてみます。（吉田功）

ザ・キャッスルのひたすら超絶ハイパーむずかし版

キャッスルエクセレント

Castle EXCELLENT

Windows　Pocket PC

1986年 発売：アスキー
作者：吉田功、岩倉啓祐

ふーう、ザ・キャッスルは難しかったけどヤリガイがあったなぁ。それどころか、終わってみると、ちょっと物足りない気分。そーんな、まだまだやる気まんまんのキミに、驚愕の続編をプレゼントしよう。それが「キャッスルエクセレント」である。マップは難しくなっているし、パズル要素も複雑だし、タイミングもシビア。こんなのクリアできるの？　一応できるハズでござる。というわけで、ザ・キャッスルの元担当者も太鼓判を押すムズカシのキャッスルエクセレント。それでもやってみるかい？

9つのブロックが行く手を阻む

前作「ザ・キャッスル」の続編が「キャッスルエクセレント」だ。ザ・キャッスルは、ガンバリさえすれば最後まで行けるぐらいの難易度だったが、本作品は作者（移植者）の岩倉氏が腕にヨリをかけて（かけすぎて）作ってしまったモノ。前提として、少なくともザ・キャッスルをクリアしたぐらいの経験がないとお話にならないぞ。

ゲームのルールについてはザ・キャッスルとまったく同じだ。登場するアイテムにも違いはない(赤レンガの色が違う……なんて指摘した人は、なかなかやり込んでるね)。純粋に、マップとアイテムのレイアウトだけが違っていると考えていい。したがって、ザ・キャッスルで身につけたテクニックとタイミングの勘は、ここでもそのまま活かせる。今再びこの難関に挑むのだ。

第1ブロックを攻略せよ

キャッスルエクセレントのマップは、ザ・キャッスルと比べて違っているところがある。それは、全体が9のブロック(エリア)に美しく分かれていることだ。したがって、1ブロックずつ着実に攻略していくのが基本。実際には、次のブロックに行くには"あること"をしなければならない。それは妖精を助け出すことかもしれないし紫の鍵を取ることかもしれない。できるだけ自分で考えてみてほしい。ちなみに第1ブロックを終えるには青い鍵が必要で、青い鍵を取るには緑の鍵を5つ集めなければならない。そして緑の鍵を取るには、さまざまなパズルを解いたり、アクションをこなさなければならないというわけで、うーん、これはなんだか大変そうである。なお、適当な順番で部屋を進んで鍵を取ったり、あるいは壺などのアイテムを壊してしまうと袋小路になり、いわゆる「詰み」の状態になってしまうこともある。このときには素直にあきらめて最初からやり直すしかない。こういうときのためにも、ブロックをクリアしたときには忘れず途中セーブをしておこうね。

ゲーム解説

キャッスルエクセレントの操作

キー	意味
[←][→]キー	左右に動く
[スペース]キー	ジャンプ
[CTRL]キー	速度アップ
[CTRL]+[GRAPH]キー	速度アップ(大)
[F1]キー	自殺
[F2]キー	ゲームオーバー
[F3]キー	データ読み込み
[F4]キー	データ保存

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

キャッスルエクセレントのスタートはここからだ。ドアを開けると、いきなり騎士がやってくるから飛び越そう。あのマップ取りて～。

最初のブロックを越えるために必要な青い鍵はここにあるのだが、緑の扉が4つも立ちふさがっているのだ。先が思いやられる今日この頃。

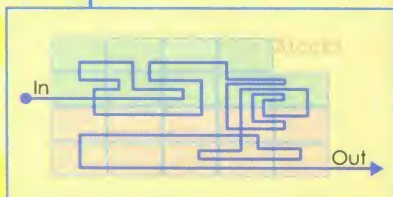
上からブロックがワラワラと落ちてくる面。最終的にこの形にすれば上に登ることができるのだが、どうやって積み上げたらいいものか。

ザ・キャッスルでは取れたチェリーが、キャッスルエクセレントでは、完璧に封印されているのだ。甘えは許さないというメッセージ？

one point column ワンポイントコラム

スーパーヒントブックを見よ！

キャッスルエクセレントをど～しても解きたいキミのために、コレ以上のものはないプレゼントを用意した。それが「キャッスルエクセレント・スーパーヒントブック」だ。これは、キャッスルエクセレントとほぼ同時期にアスキーが発売した攻略本だ（著者は、当時アスキーでキャッスル担当の新保氏）。このPDFデータを本誌付属CD-ROMに入れておいたので各ブロックごとのルート（道順）だけでも見ておこう。もちろん自分だけの力で解いたほうが立派だけどね。

スーパーヒントブックに掲載されているブロック1の攻略ルートをピックアップしてみた。適当なルートで進んでしまうと袋小路になってしまうのだ。

動かせるアイテム

ローソク

メラメラと燃えているローソクだ。炎の部分に接触すると王子は焼けて死んでしまう。押すと場所を移動できる。

コントローラ

ゴンドラを動かすためのコントローラ。ゴンドラはコントローラを左に押せば上がり、右に押せば下がるのだ。

壺、樽、レンガ、金庫

上に移動するときの踏み台にしたり敵キャラを潰すための武器として使える。ずらして重ねると階段みたいにも使えるぞ。

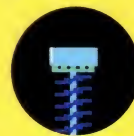
主な設備

ベルトコンベア

ここに乗ると自動的に左右に運ばれてしまう。これは動かせるアイテムも同じだ。逆方向に移動するならジャンプ！

エレベータ大

上下に運んでくれる。天井がなければ上の階へも行ける。気づきにくいのだがエレベータの下の穴は通り抜けが可能だ。

エレベータ小

上下に運んでくれる。天井がなければ上の階へ行けるものの、下の階へは行けない。幅は狭いけどやっぱり楽ちん。

バリア

一定間隔で螺旋型光線が動いたり止まったりする。止まっているときには下へ落ちるし、上へ通り抜けもできる。

不死身光線

この光線を浴びると一定時間は敵キャラに触れても平気な「不死身の身体」になるのだ。それってすごいジャ～ン。

針

横に出ていたり、上や下から出ていたりと、いろいろなパターンで生えている針。触ると死んでしまうのだ。あイタタ。

ブロック

空を飛んでいるブロック。上に乗ると左右に運んでくれる。運ばれるときには同方向に一緒に歩かないと落ちちゃう。

ゴンドラ

コントローラとケーブルで接続されていて、それを押して動すと上下する。物を運ぶのに使う人力エレベータですな。

作者から一言

「ザ・キャッスル」に最初に関わったのは、パソコン版の「キャッスルエクセレント」のマップ作成でした。当時はパズルアクション作品がポチポチ現れ始めたばかりでしたが、このゲームの潜在的な奥深さは相当なものだと思いました。そして、パズル作成者として、どこまで難易度を上げられるかという実験がパソコン版の「キャッスルエクセレント」でした。その後MSX版のお話をいただいたとき、プログラムも担当させてほしいと頼んだようにも記憶しています。MSXという恐ろしく制約のきついハードにクオリティを落とさず移植するという作業そのものが、とてもアクロバティックでパズルじみていたものでした。少ない色数を多く見せる工夫、狭い画面を感じさせない工夫、少ないワークメモリで100面を動作させる工夫、マップに注ぎ込む新しいアイデアの数々、どれもが過去に自分が関わった仕事の中でも最高レベルの充実度でした。このゲームをプレイしながら、そんな製作者側の思い入れも感じて頂けると嬉しく思います。（岩倉啓祐）

これぞゲーム界伝説の団体行動的RPG

ボコスカウォーズ

BOKOSUKA WARS

Windows | Pocket PC

1984年 発売：アスキー
作者：ラショウ

時は中世、バサム帝国の暴君オゴレスは世界征服をたくらみ、隣国のスレン王国に攻め入った。いくたびかの戦乱はあったが、バサム帝国の強大な軍事力にかなうはずもなかった。しかし、スレン王国もついに反撃するチャンスに恵まれることに。スレン王は10人の騎士と、14人の兵卒を連れ、バサム城に向かって突撃を開始した。牢屋に閉じ込められている兵卒を助けると味方は総勢50人。これだけの兵力がキミの指先1本に託されたのだ。敵国バサム帝国を倒せる日はくるのか。

進め進めものど〜も

8ビットパソコン用ゲームの中でこれほど人を魅了したゲームがあっただろうか、と言いたい人がゴマンといるはずの、不思議な魅力をもった団体行動的RPGだ。世界征服を企むバサム帝国に対してスレン王国が起死回生の反撃に出る、というストーリーからすると一種のリアルタイム戦術シミュレーションゲームといっていい。ともかくキミの指先1本で操れる味方は総勢50人。彼らを率いてバサム帝国までの600メートルの道程をひたすらズンズン突き進み、最終的に暴君オゴレスを見事に倒せば勝利である。逆にスレン王が倒されてしまったときには負けになり、残念ながらそこでゲームオーバーとなってしまうのだ。ちなみに勝利したときには次のレベルに進むことができる。そのときには敵国の兵の強さが10ずつ上がっているからより大変だ。レベルは全部で5段階存在するぞ。

じゃまな敵をけちらせ

操作方法だが、矢印キーを押すとスレン王を中心に、騎士、兵卒がまとめて動く。したがって、これで全員を左方向に導いてやればよい。また、[スペース]キーを押すたびに、動かす対象が、全員→王→騎士→兵卒の順に切り替わる(画面上に選択が表示されるので確認しよう)。これで特定の種類の兵だけを動かすことも可能だ。最初動かすことができるのは、王の他には10人の騎士と14人の兵卒のみ。途中、牢屋に囚われている兵卒を助けることで味方を増やすことができる。敵は、同じエリア内にスレン王がいると活動して近づいてきたりする。敵が現れたらその方向に体当たりすれば戦闘開始だ。勝敗は、互いの強さと、そのときの運によって決定される。負ければ死んでしまい、勝てば強さが10ずつ高くなる。騎士と兵卒は敵に連勝すると重騎士や重兵卒にレベルアップするぞ。

ゲーム解説

ボコスカウォーズの操作

キー	意味
[↑][↓][←][→]キー	上下左右に動く
[スペース]キー	移動対象の切り替え

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

進め敵の城へ〜♪

兵を動かすと一人二人岩や木にひっかかり置いてけぼりになることがある。そんなときはスレン王だけ引き返して連れてくるとよい。なお、道端の森や大木、そして領域を隔てる壁はスレン王が消せるし、兵卒が捕らえられている牢屋は騎士が消せる。また、レベル2以上で登場する"呪いの落とし穴"は兵卒が消せるので、こうしたキャラ単位の能力も大いに役立てたい。敵はこちらが動かなければ、一切攻撃をしかけてくることはない。したがって、落ちついてゆっくりゲームを進めることが肝心。慌てて進んでも即全滅だ。特に隊列はできるだけ乱さないようにしたいところ。一部の兵が先に進んでしまって確認できないところで戦わせてしまうと、そのまま無駄死にさせてしまう可能性も高い。一歩一歩じっくり進めないと最後まではいけないぞよ。

スレン王国側の登場キャラ

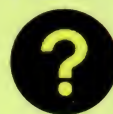
スレン王（強さ220〜320）
スレン国の王様。王が倒されるとゲームオーバーだ。王は森の大木や領域を隔てる壁を消すことができる。

騎士（強さ150〜310）
4連勝すると重騎士に昇格し、さらに強さが80増える。騎士は兵卒が捕われている牢屋を破ることが可能。

兵卒（強さ30〜310）
3連勝すると重兵卒に昇格し、さらに強さが80増える。人質として捕らえられている兵卒は助けてやろう。

バサム帝国側の登場キャラ

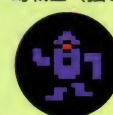
オゴレス王（強さ100）
オゴレス王を倒せばレベルクリアとなる。しかし、オゴレス王をスレン王以外で倒すのは、非常に難しい。

騎士（強さ50）
最初の草原、次の森林、そしてバサム城の手前の草原に出現する。強さは中の下ぐらいだが油断は禁物。

親衛隊（強さ250）
バサム城に登場する、もっとも強い敵。しかし、実は兵卒に弱く、重兵卒に勝つことはほとんどない。

兵卒（強さ10）
騎士と同じく、草原、森林、荒野に登場する。弱い敵なので、意図的に味方兵をぶつけて昇格させよう。

幻戦士（強さ50）
倒してもすぐに復活し、スレン王をしつこく追いまわすイヤな敵。主人の魔術師を倒すと消滅してしまう。

重騎士（強さ100）
城跡とバサム城に登場する強敵だ。スレン王をしつこく追いまわしてくるので、すり抜けるときは注意。

魔術師（強さ30）
岩場とバサム城に登場する。スレン王がわきを通るときに、魔術師は幻戦士を召喚して呼び出してくる。

獄師（強さ30）
牢屋の番人。牢屋付近で捕虜となっているスレン軍の兵卒を見張っているので、倒して兵卒を助け出せ。

レンガに囲まれた魔術師のわきをスレン王が通過しようとすると、4人の幻戦士が現れる。こいつを倒すには魔術師を攻撃だ。

領域の近くには兵卒が牢屋に囚われていたりする。騎士だけ動かして牢屋を消して助けよう。また、領域の壁は王を動かして消していくこと。

重騎士とはできるだけ戦わせずやり過ごすのがコツ。重騎士はスレン王の一歩手前までしか進まないので、スレン王を縦に動かして重騎士をおびきよせる。そして、味方を横一列に並べて、そのうしろをスレン王に通り抜けさせるのだ。

無念。スレン王が死んでしまった。この世に正義はないものか。

one point column ワンポイントコラム

オゴレス倒すのだぁ!!

すぐ死んでしまうという人のためにヒントを授けよう。まず、敵にうしろから体当たりして、戦闘中もうしろむきに動かし続けると勝率が高くなる。同じ種類の兵隊が後退するという副作用はあるが、これは覚えておきたい。この技と魔術師を利用すると兵隊を鍛えることができる。まず鍛えたい兵隊だけを魔術師のうしろに移動したら、王を進めて幻戦士を呼び出す。そして、背後から戦いをしかけるのだ。王を強くするには敵兵卒とじっくり戦わせよう。最高の強さ（320）までいったら、以後王は戦わせず最後まで温存すべし。さあこれでオゴレスと最終決戦だ！

ついに来たバサム城。数々の難関を乗り越えてここまでたどり着いたからには、なんとかオゴレスを倒したいーっ！

作者から一言

今でこそ謎のソフトハウス「イタチョコシステム」の総帥として、そして奇跡の天才ゲームデザイナとして不動の地位な私であるが（ウソ）、その当時は夢多き、いや夢が多過ぎて交錯し、大学なんかメンドくさくて行ってらんない青少年であったのだ。その創作意欲のテンコ盛りをまとめることができたのが「ゲーム」というジャンルであったのだ。正確にはゲームがそんなジャンルじゃないかと思っちゃったのだ。世間にゲームの可能性を問うという大風呂敷も広々に、しかし結局クソゲーと語り継がれることとなるボコスカウォーズなのであった……。普通の人間としての道を外した記念すべき作品である。後悔してはいませんよ……あんまり……。（ラショウ）

自動照準がカッコよすぎの変形メカアクション

テグザー

THEXDER

Windows　Pocket PC

1986 ©GAME ARTS

宇宙暦089年。宇宙兵器開発船レイピナは、後にネディアムと名づけられる謎の小惑星を発見。探査のために数万キロまで接近したところ、突然小惑星の強力な磁場に捕らえられ脱出不可能になってしまった。そこで乗組員たちは急遽開発中の新型兵器ハイパーデュアルアーマー「テグザー」の始動を決定。状況に応じて2タイプに変形可能なテグザーは、あらゆる局面を打開できる性能をもっている。キミはテグザーを使って惑星内に侵入し、磁場を形成している根源を破壊しなければならない。成功を祈る！

いかす変形ロボットゲーム

新型兵器ハイパーデュアルアーマー「テグザー」を操作して、謎の小惑星「ネディアム」の地下要塞に侵入し、宇宙兵器開発船「レイピナ」を捕らえている強力な磁場の発生源を爆破するのが本ゲームの目的だ。迷路状の各ステージには、グループでおそってきたり、後ろからしつこく追いかけてきたりと、多種多様な敵が待ち受ける。これらを、ときには倒し、ときにはやりすごして、どんどん右方向に進んでいかねばならない。先に続くステージは全部で10面。ステージによって雰囲気も敵の種類もどんどん変わっていくぞ。むろん戦闘力もアップしてくるから大変だ。なお、1つのステージ内では自由に戻ることが可能だが、先のステージに進んでしまうと、シャッターが下りて、前のステージに戻ることはできなくなる。したがって、やるべきことは全部やってから次に向かおう。

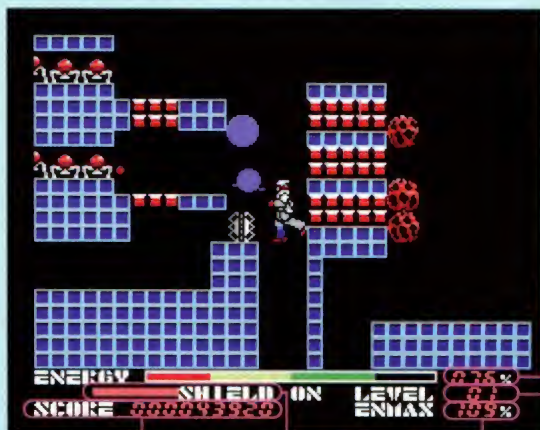
気持ちいい自動照準ビーム

テグザーにはロボット形態と飛行形態があって、簡単に切り替えることができる。ロボット→飛行に変形したいときには[↓]キーを押し、飛行→ロボットに変形したいときには進行方向と逆向きの矢印キーを押すか、どこかの壁にぶつけよう。

ロボット形態では、地面を走って進むことになる。飛行形態と同じぐらいの速度で走れるし、かなり高くジャンプもできるので、機動性は抜群だ。逆に飛行形態ならば、高いところまでいけるし、狭い通路に入っていくこともできる。これらは状況によって使い分ける必要があるのだ（ちなみに、大ヒントだがレベル4の出口にいるジューダスという敵は、飛行形態でやっつけるといいぞ）。

敵を倒すために備わっているテグザーの武器はビームライフル。[スペース]キーを押せば一瞬で攻撃だ。飛行形態時には真正面に放たれるし、ロボット形態時には自動照準が働いて近くの敵をくまなく一掃してくれる。ただし、硬い敵に対しては、ビームライフルを当ててもすぐに死ぬわけではなく、かなり時間がかかるから辛抱強く当て続けること。

テグザーには、敵からの攻撃を防いでくれるシールドも備わっている。これは[SHIFT]キーを押すことで作動する。そのまま何事もなくてもシールドはジワジワ減少してしまうし、敵の攻撃を防御したときには急激に減少する。シールドは残量が0%（OFF）になるまではダメージはない。しかし解けた瞬間からダメージがくるから、シールド残量がなくなってきたら再度[SHIFT]キーを押して新しくシールドを張り直しておこう。

ゲーム解説

テグザーの操作

形態	キー	意味
ロボット形態時	[←][→]キー	左右へ移動する
	[↑]キー	ジャンプする
	[↓]キー	飛行形態に変形する
飛行形態時	[↑][↓][←][→]キー	上下左右へ移動
共通	[スペース]キー	ビームライフルを撃つ
	[SHIFT]キー	シールドを張る
	[ESC]＋[Q]キー	自爆

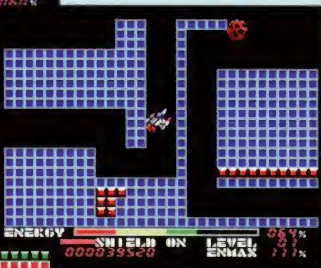
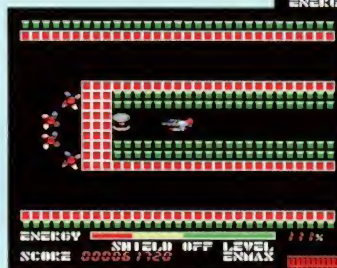
注　ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンAに、[SHIFT]キーがトリガーボタンBに対応する

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

エネルギーはいつも貴重品

テグザーは、走ったり飛行しているだけならエネルギーは使わないのだが、ビームライフルを使うとその分だけ少しずつ消費してしまう。したがってむやみやたらに撃てばよいというものではないのだ。無駄使いはせずに、ちゃんと敵が近くにきたときに狙おう。また、シールドを張ると、その瞬間に10%減る。さらに、敵とじかに接触してしまうと、またたくまに減ってしまう。こうして、最終的に0%になってしまうと全機能停止で自爆...ゲームオーバーである。したがって、テグザー攻略にはエネルギーの維持をどうするかが最大のポイントとなってくる。エネルギーを補給するには特定の敵を破壊すればよい。敵によってはエネルギー総容量を少し増やしてくれたりもするので、先のステージでの体力をつけるためにも、これらの敵はなるべく逃さないようにしたいところだ。ちなみにエネルギー総容量はステージクリアするたびに10%増加される。あとはキミの腕次第だぞ。

ロボット形態でエネルギーかもらえる敵を破壊。[スペース]キーを押しているだけで、まとめて攻撃してくれるぞ。この武器が当時衝撃的だった。

こういうミゾにハマってしまったときには[↓]キーを押した後で、すぐ[↑]キーを押す。これで飛行形態に変形後すぐ上昇できるのだ。

ステージ2になると強い敵もうようよ。こんな隙間にあるエネルギーもちゃんと補給しておきたい。だけど飛行形態でじゃないと入れない。

ステージ3のラスト。この先左にはエネルギー補給キャラが多数並んでいるが、シャッターが邪魔で進めない。この位置でシャッターを破壊だ。

テグザー2形態

ロボット形態

空は飛べないが、走ったりジャンプは得意。勝手に攻撃する自動照準のビームライフルがかっこいい。

飛行形態

高い場所に行くならこちら。狭い隙間にも入っていける。ビームライフルは進行方向のみに発射される。

エネルギー補給キャラ

これらのキャラを破壊するとエネルギーが補給されたりエネルギー総容量を増やしてくれたりするので積極的に取ろう。

エネルギーA

空中に静止していて破壊するとエネルギーが増加。エネルギー総容量はそのまま。

エネルギーB

地面で静止していて破壊するとエネルギーが増加。エネルギー総容量が2%増える。

エネルギーC

地面を動いている敵で破壊するとエネルギー総容量を1%だけ増やしてくれるのだ。

one point column ワンポイントコラム

ごめんね人形はいずこ

ビームライフルを撃っていると壁の中から敵が現れることがある。これが隠しキャラだ。見つかるとこちらに縦座標を合わせて動いてくるのだ。これを破壊するとエネルギーを回復してくれる。どれも硬いけれどなんとかやっつけたい。隠しキャラはだいたいステージの最後のエリアに隠れていることが多い。見つからない人は、その辺りを自動照準のビームライフルで撃ってみよう。何もない方向に反応したら、そこに隠しキャラがいるかも...よ。

ステージ1～3の隠しキャラ。一番右の「ごめんね人形」が超懐かしい。

俺とテグザー

テグザーは最初NECのPC-8801SR専用ゲームとして発売され、アミューズメントゲームばりの迫力と、当時流行りだったロボットアニメのモチーフと、自動照準の斬新さなどがあいまって、多くの人に支持されたタイトルだ。そんなテグザーがMSXに移植されたときにもやっぱり話題で、当時のMマガのレビューページでも「おすすめマークほい！」で初の5つ星をゲットしているのだ。今回収録できて心底よかったなぁ。そうそう、忘れちゃいけないけど、テグザーはBGMもイカしている。トンテントンテン...あーテグザーって感じ。ラストの「月光」も寂しくてグー。ちゃんと聴くべし！

新アイテムを引っさげてあの変形メカが帰ってきた

ファイアーホーク

FIRE HAWK

Windows　Pocket PC　1989 ©GAME ARTS

テグザーの働きで磁場が弱り、小惑星ネディアムから脱出できたレイピナだったが、搭乗していたアーサー大尉もろともテグザーをロスト。さらに不思議なことに、事後の調査では小惑星の存在そのものが確認できなかったという。そして4年後、あのネディアムが再び姿を現した。しかもそこは太陽系で、地球に向かっているという。このままでは衝突はまぬがれない。回避の方法はただ1つ。テグザーに代わって新造された「ファイアーホーク」部隊によってネディアムを破壊するのだ。

うんと情報収集するでありマス

本作はテグザーの続編だ。再び姿を現し、あろうことか地球に向かってきた小惑星ネディアム内部に、テグザーの4倍のスケールと武装強化された「ファイアーホーク」で突撃。ネディアムを破壊するのが目的だ。搭乗するのは「ジョシュア」。テグザーのパイロット「アーサー大尉」のフィアンセだった女性だ。本ゲームによって、ネディアムの謎、そしてアーサー大尉の行方が解明される。こうしたストーリー性が、いっそうゲームに深みをもたせているというわけだ。ゲームを開始するとそこはネディアムの地表。例によってさまざまな攻撃をしてくる敵をかいくぐり地下へと進行してゆく。ところどころにある通信ポッドや、先行し撃墜されたファイアーホークの残骸を発見したら[CTRL]キーでメニューを表示して、CONTACTを選んでさまざまな情報を収集しよう。

ファイアーホーク2形態

ロボット形態

ちょっとした足場があれば着地やジャンプが行える機動性。自動照準のレーザーが最高なのだ。

飛行形態

狭い場所でも垂直上昇、垂直下降が自由自在。レーザーは進行方向のみに発射されるぞ。

サブウェポンを選ぶでありマス

操作はおおむねテグザーと同じ。ロボット形態と飛行形態の変形もできるし、テグザーシリーズの最大のウリとなっている自動照準のレーザーも健在だ。そこに、本作ではホーミングミサイルと、その他のサブウェポンが追加されている。これらの兵器はすべてステージ上に落ちているアイテムを拾うと補給できる。使用するには[CTRL]キーでメニューを表示してSHIELD、MISSILE、OPTIONのうちいずれかを[↑][↓]キーで選択し、最後に[スペース]キーで確定。あとは[SHIFT]キー(前作でシールド専用だったキー)で使用できる。なお、サブウェポンは最大3つまで保有でき、OPTIONの

ゲーム解説

ファイアーホークの操作

形態	キー	意味
ロボット形態時	[←][→]キー	左右へ移動する
	[↑]キー	ジャンプする
	[↓]キー	飛行形態に変形する
飛行形態時	[↑][↓][←][→]キー	上下左右へ移動
共通	[スペース]キー	レーザー発射
	[SHIFT]キー	シールドを張る
	[CTRL]キー	メニューを出す

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンAに、[SHIFT]キーがトリガーボタンBに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでトップメニューが表示される。[STRAT]を選択して[スペース]キーを押すとミッションが表示されるので、再度[スペース]キーを押すとゲームが開始となる。

メニュー	プレイ中[CTRL]キーを押すと表示される。その間は画面が静止するので慌てず選ぼう。
CONTACT	通信する　通信ポッドやファイアーホークの残骸近くで使おう
SHIELD	シールドを選択　以後[SHIFT]キーでシールドがオンになる
MISSILE	ミサイルを選択　以後[SHIFT]キーでミサイルが発射される
OPTION	サブウェポンを選択　以後[SHIFT]キーで兵器が使用される

ミッション1のスタート。まずはネディアムの地表に降り立ったファイアーホーク。ちらほら出てくる敵に注意しながら、とにかく右方向へ進もう。

最初に出てくるボスキャラ。赤い玉ころが壁で跳ね返りながら襲ってくるのだ。ついでに上中下からは小弾を連発。画面の左に寄って少しずつ落とすのだ。

おや、ここはどこかで見たことがあるなと思ったら、テグザーのステージ1と同じフロアー。でも前回の戦闘のせいかあちこちが崩れてしまっているのだ。

地中奥深く、うねった道の進んだ先に、出ましたミッション2のボスキャラ。中央を狙うらしいのだがどう対応すればいいのか。オタオタしてたらやられちゃうぞ。

メニューからさらに選択する形になっている。ただし、ストッパーとECM作動中は選択ができないので要注意。また使用回数は兵器によって異なっているので、画面左下で確認すること。

ボスはとても強いでありマス

ファイアーホークはテグザー同様にエネルギー総容量には限界がある。レーザーを使うと少しずつ消費するし、シールドを使うと10％ずつ減っていく。そして当然、敵に接触すると著しく減少する。エネルギーを補給するには特定の敵を破壊すればよい。ステージによって補給できる敵の形が異なるので、そこはしっかり見極めたい。とはいえ、たいていは静止している"いかにも"なキャラがそれなので心配は無用だ。ともかくエネルギーが0％になってしまうとゲームオーバーなので補給を常に心がけること。

本作はミッションという単位でシナリオが進んでいくが、その終わりに明確なボス対決シーンが用意されている。それぞれの弱点や注意点は、途中の通信ポッドなどを使ってコンタクトし、情報を地道に収集しておこう。でないと、無駄な攻撃をえんえん続けることになって、消耗の末にオダブツとなっちゃうぞ。地球の存亡を賭けてふんばるのだ！

one point column　ワンポイントコラム

プラクティスモードで練習せよ！

起動時のメニューで「PRACTICE」を選択すると練習モードになる。ここでは攻撃されてもダメージはないので安心して使い方を覚えよう。しかし練習とはいえ、その内容はかなり過酷。これから先の地獄を見るようで恐ろしい。もっとも、アイテムなどを使って先へ進めることを覚えたりすると、かえってやる気が出ちゃうかも。なお、詳しくは下ぶくれの教官が言葉で説明してくれるので、通信ポッドの前で[CTRL]キーを押してコンタクト（CONTACT）してみて。

「ホーミングミサイルに切り替えておくんだ」と教えてもらったからには、いっちょやってみるであります。

俺とファイアーホーク

テグザーはかなりやり込んだけどファイアーホークは初めてという人多いのではないかな？ところが実際やってみると最初はごつごつした岩肌ばかり。テグザーのスマートなマップとの落差で、いまいちノリが悪い人もいることだろう。そういう人は、がんばってともかくミッション2まで進んでみてほしい。そこに懐かしいテグザーのステージ1が登場するのだ。しかも、少しばかり朽ち果てていてアンニュイな気分。このあたりでグッと感情移入しちゃうぞ。30才代前後のキミの青春ってやつがソコにある…たぶん…きっと。

アイテム

ミサイルポッド
取るとホーミングミサイルを30発ずつ補給してくれる

エネルギージェネレーター
取るとエネルギー総容量を最大30％増やしてくれる

サブウェポン

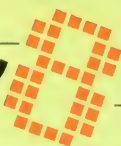
エネルギータンク
選んで使うとエネルギーを1回だけ満タンにしてくれる

ストッパー
選んで使うと敵の動きを一定時間だけ止めてくれる

フラッシャー
選んで使うと画面上のすべての敵に一定のダメージを与える

ナパーム爆弾
選んで使うと炎が地面にいる敵を焼き払ってくれる

ECM
選んで使うと敵の弾による攻撃をピタリと停止してくれる

ダートミサイル
選んで使うと進行方向に強力なダートミサイルを発射する

今世紀最大のゲーム大会にリベンジ

ファミクルパロディック2

FAMICLE PARODIC2

Windows　Pocket PC　1990 ©BIT²

いつものごとくゲームに熱中していたファミクル一家に1枚のチラシが舞い込んだ。「ゲームハウスパロディアンにてゲーム大会を行います。各自マシン持参で参加してください。優勝者には宇宙旅行をプレゼント！　ぜひご参加を」。これは黙っていられないファミクル一家。さっそく各自でマシンを製作することに。大会当日はたぁーくさんの参加者が集合。アキレス腱を伸ばす者、マシンにオイルを差す者、ブルワーカーをする者など、みな緊張をほぐそうとしているぞ。なんとかマシンを作り上げたファミクル一家もスタートを待つばかり。さて、大会の行方は？

パローれパロれパローれ

ファミクルパロディック2は、前作「ファミクルパロディック」の第2弾として発売された縦シューティングだ。BIT²ファンならお馴染みのファミクル一家が大活躍。家族の一人を選んで、スクロールするステージの上から次々と現れてくる敵をやっつけていけばよい。とはいえ、途中には中ボスも登場するし、最後には大ボスも登場。それぞれ変化にとんだ攻撃をしてくるぞ。さて、本ゲームが楽しめるもう一つの要素が、タイトルにもあるように「パロディ」の部分。歴史に残るゲームの風味がそこはかとなく漂うマップ。どこかで見たような感じのボスキャラ。どこかで聞いた音楽フレーズ。ゲームが大好きな人なら、新しいステージに進むたびに、ついついニンマリしちゃうのだ。

攻略パターンを構築せよ

シューティングゲームだけに、基本は撃ちまくり（押しっぱなしで連射せよ）避けまくりなのだが、敵の攻撃はトリッキーだし、地上からは休みなく攻撃されるし、道は狭いしで少し進むのも大変だ。このあたりは第2弾のゲームということで難易度も高くなっているのだ。ちなみに、途中でやられてしまうと、ステージの最初か、もしくは中ボスを倒した後からのスタートになる。一気に中ボスおよび大ボスを倒さないと、かな～り疲労こんぱいだ。ただし、最初5つある自機が全部やられてしまってもコンティニューができるので、不屈の闘志で何度でもチャレンジしてほしい。自分で少しずつ攻略パターンを構築していって、神ワザのような操作で友達を大いに唸らすのだ。

オプションを使いこなせ

本ゲームを攻略するのに、どうしても習得しておきたいのがオプションの使いこなしだ。

ゲーム解説

ファミクルパロディック2の操作

キー	意味
[↑][↓][←][→]キー	上下左右へ移動
[スペース]キー	弾を撃つ
[GRAPH]キー	オプションの弾の方向を変更
[スペース]を押しながら[GRAPH]キー	オプションの遠近を切り替える
[F1]キー	一時停止

注：ジョイスティックでは、矢印キーかレバーに、[スペース]キーがトリガーボタンAに、[GRAPH]キーがトリガーボタンBに対応する。

起動時の操作　ゲームを起動して3回[スペース]キーを押すとキャラクタ選択モードになる。ここで操作するマシンを選ぶ。[スペース]キーを押すとゲーム開始だ。

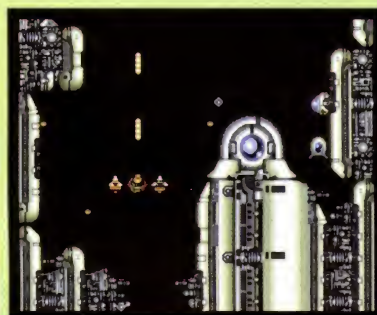
パワーアップアイテム

オプションが増えたり主砲が強力になるのだ。

なんだか懐かしい色合いと地上絵。ファーストステージはもちろんゼビウス風味なのだ。ピラミッドや地上絵もあるよ

ステージ3はファンタジーゾーンみたいな雰囲気のステージだ。しかも敵を倒すとコインになったりもする芸の細かさ。

ステージ1の大ボス、アンドアジェネシスみたいに多数の弾を撃ってくる。中央のコアを粉砕するまで避けて撃つべし。

ステージ3の大ボスは両手からタコを投げてくる。それを撃墜しながら出ベソを狙おう。これってもしやパロディウス?

ステージ2は、R-TYPEもしくはグラディウスふう。オプションを使いこなさないと進めないぞ。

登場キャラ

パビクル
パパの乗るアンコウ型マシン。大きなウェーブ系のビームを装備しており広範囲を攻撃。オプションからは弾が斜め上に1つ発射されるが左右クロス方向なのユニーク

マミクル
ママの乗るカタツムリ型マシン。レーザー系のビームを2門装備しており比較的当てやすい。オプションからは弾が1つ発射。斜め前→横→斜め後方向に切り替わる

ミロクル
息子が乗るカエル型マシン。バブル系のビームを装備。パワーアップすると2門幅になるのが嬉しい。オプションからは弾が1つ発射。前→横→後方向に切り替わる

シンクル
娘が乗るアヒル型マシン。レーザー系のビームを装備。オプションからは弾が2つ発射。斜め前+前→斜め前+斜め後→斜め後+後という組み合わせで切り替わる

ニャンクル
猫が乗るネコ型マシン。ウェーブ系のビームを装備しているが幅が小さい。オプションからは弾が2つ発射。前+横→前+後→後+横という組み合わせで切り替わる

ステージ上のアイテムを取るとオプションが最大2つ装備され、そこからも弾が発射される。しかも、[GRPAH]キーを押すたびに、弾の発射方向(フォーメーション)を3段階で切り替えることができるのだ。さらに、[スペース]キーを押しながら[GRPAH]キーを押すことで、オプションの遠近を切り替えることもできる(オプションが遠いときにも[GRPAH]キーで発射方向を切り替えることが可能だ)。これを利用すると、壁を隔てた向こうの敵を攻撃することも可能となる。主砲が届かないところにいる敵は、これでバシバシやっつけよう。

one point column　ワンポイントコラム

ディスクBでDEMO鑑賞だ

ファミクルパロディック2はディスク版のゲームだ。MSXPLAYerのスキンを見ると、ディスクイメージAとディスクイメージBが表示されていることからも明白だ。そこで試しにMSXPLAYerでディスクイメージBを選択してからリセットしてみてほしい。するとどうだろう、本ゲームのDEMOが表示されるのだ。見てみるとファミクル一家がどんなキャラなのか。誰がどんなマシンに乗っているのかがよーくわかる。これで世界観をつかんでちょ!

これはニャンクル。ニャンクルだけに猫なのだ。乗り物も猫型ニャン。マシンの名前はスィエルシャート号って言うんだにゃ。

俺とファミクルパロディック2

ステージ1から激ムズのこのゲーム。攻略のポイントとなるのはキャラ選択だ。マシンによってビームの当てやすさが違うし、オプションの役立ち度も異なるから、すべて試してチョイスしたい。お勧めはパビクルとマミクル。斜め前方に攻撃できるオプションが便利。特に、最初はステージ1の中ボス攻略が大変なのだが、この場合でも、迫る壁に潰されないように注意しつつ、オプションの斜め攻撃で左右の像を攻撃できる。これで楽勝!その先は神のみぞ知る…だニャンニャン。

クイ打ち名人が挑むアクションパズル

クインプル

QUINPL

Windows　Pocket PC　1988 ©BIT²

むかーしむかし平和を愛する神様がおったとさ。しかし、平和になりすぎて神様はなぁーんにもすることがなくなってしまったのだ。そこで神様は「はっ！」とひらめいた。「いたずら好きのクインプルとツウィンプルをこらしめてやろう」。そう思った神様は2人を広い迷宮に閉じ込めてしまったのだ。当の2人は何が何だかわからない様子。キョトンとして辺りを見回してみると空から神様の声が。「時間内にこの迷宮から脱出するのじゃ。ヒヒヒッ。もし脱出できないのならカエルにしてしまうぞ。オホホホ」。さぁて2人は迷宮脱出ができるだろうか。

に　クイ難解パズルだね

手にしたクイ（杭）を使って、全100面ものステージをクリアしなければならないのがクインプルだ。ルールは簡単。主人公たるクインプルを、ブロックやブロックに刺したクイに乗ったりして、ともかく出口まで誘導してやればいい。ただし、クインプルには制限時間があるから行動は素早くね。ステージをクリアすると、性格の悪い神様のありがたくな〜いお言葉がいただけるぞ。神様の期待を裏切ってどんどん攻略していこう。逆に、最初5人いるクインプルをすべて失ってしまうと、残念ながらゲームオーバーだ。神様がコンティニューするかどうか尋ねてくるから、するなら[GRAPH]キーかトリガーボタンBを押そう。これで失敗したステージからプレイできるぞ。

転　ばぬ先のクイだね

クインプルを制覇するには、クイの使い方を完全マスターすることが必須だ。こいつを極めないとステージ1から立ち往生になってしまうぞ。まず、[スペース]キーを押すとクイが横に発射されるが、[↑]や[↓]キーを押しながらだと天井や床にも刺せる。さらにジャンプしながらの発射も問題ない。それと、クイは放物線を描いて飛んでいく。そのため、ブロックとの距離を調節することで、やや上に刺したり、やや下に刺したりといった調節ができる。クイがブロックに当たったときにはたいてい刺さる（刺さらないブロックもある）。このとき、横と下に刺さったクイの上にはぴょんと乗れる。上に登りたいときにはこれを足場にしてジャンプしよう（ちなみに、クインプルのジャンプは長押しでより高く飛べるぞ）。ブロックに刺さったクイの上に乗ったとき「ピキッ」とヒビが入ったときには、再度クイの上でジャンプするとブロックを崩すことができる。不要なブロックはこれで消していこう。逆に、すでにヒビの入っているブロックにクイを刺して乗ると、足場

ゲーム解説

クインプルの操作

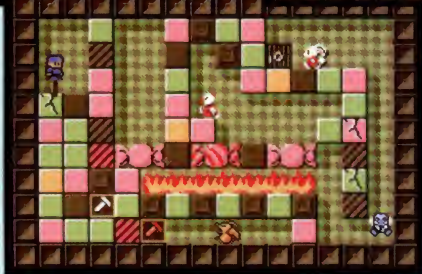
キー	意味
[←][→]（[A][D]）キー	左右へ移動する
[スペース]（[CTRL]）キー	クイを投げる
[GRAPH]（[TAB]）キー	ジャンプ
[↑][↓]（[A][D]）＋[スペース]（[CTRL]）キー	上下へクイを投げる
[GRAPH]（[TAB]）＋[スペース]（[CTRL]）キー	メニューを出す
[F1]キー	自殺

注：かっこ内は2P側のキー。ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンAに、[GRAPH]キーがトリガーボタンBに対応する。

起動時の操作　ゲームを起動して2回[スペース]キーを押すとメニューが現れる。ここではプレイヤー数が選択できる。1人でプレイするなら[1P MODE]を、2人で同時にプレイするなら[2P MODE]を選ぶ。

ステージ1ではクイの基本的な使い方を練習しよう。クイは放物線を描いて飛ぶので、離れた位置から狙えば微妙に高さ調節も可能だ。

ステージ2で必要なのは、下に打ったクイの上に乗って、ジャンプして、下のブロックを壊す技。これができないと1歩も進めないぞ。

ステージ3は縦長の形をしているぞ。ここでは、左右の壁面から3段登って、そこから内側の壁面に飛び移ってさらに登っていくのだ。

が崩れてしまって、二度と登れなくなったりもする。どうすれば先に行けるか考えよう。

クイ意地悪いお人だね

ともかくこのゲームは、どういうルートで出口までいくかが問題。そのためには、どう足場を作り、どうブロックを利用するかがカナメとなる。しかも時間に追われながらというのだから大変だ。これだけでも十分つらいのだが、さらにうとましいのが敵キャラの存在。地上をうろうろしているウサギに、空中をばたばたしているニワトリ。そして下りは早いが上りは遅い虫くん。すぐ火を投げてくる怪鳥と、せっかくルートを確立しても、アクションの段階でやられてしまう。敵キャラによってはクイでやっつけることもできるから試しに当ててみよう。なお、いい調子でどんどんステージクリアしていると10面ごとにボスキャラまで登場する。どこかに弱点があるので、そこを攻めてやっつけてほしい。

one point column ワンポイントコラム

金貨いっぱい隠し部屋

100面という途方もないステージが用意されているクインプルだが、さらに隠し面まで用意されていることが判明。そこには金貨がたくさんあって時間内ならば取り放題だ。見えない落とし穴があるため、そう簡単に全部取ることはできないが、嬉しいご褒美だ。また金貨全部を取らずとも、とにかく自キャラを時間内に出口まで誘導することにより、そのときのステージがスキップされ、次のステージが始まってくれる。1ステージ攻略しなくていいので助かるね。

隠しステージに行ける場所は決まっていて、たとえばステージ2なら中央上あたりに行くと入れる。金貨を取って時間内に出口に向かえ！

俺とクインプル

本ゲームはなかなか先に進めない。それはどうしてかを考えてみたところ、キーボードではどうも操作がしづらいと判断した。というのも、クインプルではクイ打ちとジャンプを同時にやらないと進めない箇所が初期ステージから出てくるのだが、キーボードでの同時押しはどうにもやりにくいのだ。もし、最初からイキづまっている人がいるなら、試しにジョイパッドをPCに接続してプレイしてみてほしい。ABボタンの同時押しが簡単にできるので壁昇りなども非常にやりやすい。試してみてくれ！

登場キャラとアイテム

クインプル

クイの扱いにかけては誰にも負けない主人公。実際、各ステージを攻めるには、このクイの扱い方が重要なのだよ。

ツウィンプル

2P側のプレイヤーキャラ。一応双子という設定らしくクインプルと瓜二つ。2Pプレイのときには互いの頭に乗れるぞ。

普通のクイ

最初に持っているクイ。一度に2本飛ばすことができる。2本以上連射するとクイが飛行途中で消えてしまうので注意。

黒いクイ

他のクイは投げると放物線を描いて落ちるが、このクイを使うと、ずっと一直線上に飛ばすことができる。敵でも倒す？

白いクイ

壊れるブロックに当てても壊れないクイ。どうしてもヒビを入れてしまうようなときには、このクイで足場を確保せよ。

赤いクイ

ブロックに刺すと点滅し、一定時間後に爆発。ブロックがこっぱみじんになるという、ちょっとあぶなっかしいクイ。

青いクイ

普通のクイよりも長時間上に乗っていられる丈夫なクイ。確固たる足場を用意したいときには、コイツを使うのだ。

鳥

このステージは降参だーってときに使うと、鳥だけに「1面だけ飛ばすことができる」のだ。お後がよろしいようで。

浮遊感覚満点の滑らかスクロールゲーム

テセウス

THESEUS

Windows　Pocket PC

1984年 発売：アスキー
作者：三橋正邦、滝口彰、h.godai、コミネマオ

お姫様が迷路の中に捕らえられてしまった。あなたは勇者テセウスとなり、迷路の中からお姫様を助け出すのだ。しかし、そのためには指輪と鍵が必要だし、密室に幽閉されているときには壁を開かなければならないだろう。キミの得意技というと、一度蹴ったら天まで上がれる「等速どこまでもジャンプ」。敵ロボットに触れたり放射線廃棄物貯蔵池に落ちないように気を付けながらも、この能力を最大限に活かしてガンバるのだ！　けど、生命力がなくなったら一貫の終わりっス～。

ゲーム解説

テセウスの操作

キー	意味
[←][→]キー	左右に動く
[スペース]キー	ジャンプする

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

鍵と指輪でお姫様救助

ステージのどこかでじっと助けを待っているお姫様を、プレイヤーキャラの勇者テセウスが救出するというアクションゲームだ。勇者といっても、見た目は野球帽をかぶった少年という感じだが、そこは想像で補っていただきたい。

というわけで、キミはテセウスをお姫様のところに誘導すればいいのだが、そのためにはステージ上のどこかにある「鍵」と「指輪」を入手しなければならない。手ぶらで行っても救助はできないのであしからず。そんなテセウスの行く手を阻むのが、ステージ上の敵ロボットや真っ赤な放射線廃棄物貯蔵池だ。ロボットに触れたり、池に浸かってしまうと、テセウスの生命力が急激にダウンする。そのまま生命力がゼロになったらゲームオーバーだ。

どこまでもジャンプだ

テセウスは剣も銃も持っていないのだが、非常に頼りになる武器がある。それが、物理法則を無視するかのような「等速どこまでもジャンプ」だ。[スペース]キーを押し続けると、その間はどこまでもスイイインと上昇していくから、どんなに高い場所へもひとっ飛び。まるで宇宙遊泳をしているかのような浮遊感がたんまらな～い。ただし、一度天井に頭をぶつけてしまうと、そこからは下に向かって落ちて行くのみ。高い位置から落ちてもダメージはないが、着地場所を間違えると大変だ。そこが放射線廃棄物貯蔵池だったりしたときには、残り生命力が少ないと一巻の終わりである。

OPEN/CLOSEで開いて閉じて

このゲームには時間制限があって、残り時間がゼロになると生命力が減り始める。その生命力もゼロになってしまうとやはりゲームオーバー。したがって、本ゲームを攻略するには、なるべく最短ルートで指輪と鍵を取り、お姫様の場所まで行かねばならない。ところが、そこで問題なのがOPEN／CLOSEで開閉する壁だ。OPENに触れるとどこかの壁が開き、CLOSEに触れるとどこかの壁が閉じる。初期のステージでは、基本的にはOPENだけに触れるようにすればいいのだが、後のステージになるとスイッチそのものがたくさんある上、意図的にCLOSEに触れないと進めないこともある。真面目に全13ステージをクリアしたいならマッピングするべし。

お姫様が目の前にいるのだが、壁に阻まれて会うことができない。実は、下のOPENに触れると、この壁は消滅してくれるのだ。

指輪と鍵を拾って姫のところに行けば、見事に救助することができる。それにしても大きなハート。テセウスくんの面目躍如だ。

レベル5の細い隙間。ここは非常に入りづらいのだが、ちょうど隙間の高さで速度ゼロにすると入れるのだ。テセウスの上級テクニックだ。

レベル10（ステージ10）。なんと、放射線廃棄物貯蔵池の上に指輪があるぞ。どうやったら、これを取ることができるのだろうか！

登場キャラとアイテム

テセウス

プレイヤーキャラ。テコテコと歩く姿はちょっと頼りないがジャンプ力には自信ありだ。ぐんぐん上昇する～。

指輪

お姫様のところに持っていく必要のある2つ目のアイテム。姫を解き放つのはキミの真摯な愛だけってわけだ。

敵ロボットA

四角型の敵ロボット。触れると生命力がすごい勢いで減少する。タイミングを見計らって通り抜けるのだ。

旗

何の旗なのかは不明だが、取ると900点貰えるのでスコアアップには有効。特に取らなくても問題はなさそう。

敵ロボットB

UFO型の敵ロボット。こちらも触れると生命力が激しく減少する。サイズが大きいので触れてしまいやすい。

パウダー

拾うとしばらく時間敵ロボットの動きが停止する。ただしロボットに触れると生命力が減ることについては同じだ。

お姫様

部屋の一角で勇者テセウスを待ち続けているお姫様。密室に幽閉されているときはOPENしないと会えないぞ。

開スイッチ

OPENと書かれたスイッチ。触れると対応する壁が消滅する。基本的には壁を消してゆくことで先に進めていける。

クリプトナイト結晶

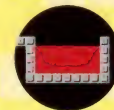
生命の源であるクリプトナイト結晶を取ると生命力が127増える。残念なことに最大999以上にはならない。

閉スイッチ

CLOSEと書かれたスイッチ。触れると対応する壁が復活する。OPENの近くにあると間違って触れてしまうことも。

鍵

お姫様のところに持っていく必要のある1つ目のアイテム。姫の閉ざされた心の扉を開くためのアイテムかな？

放射線廃棄物貯蔵池

毒々しい赤色の池。ここに落ちると生命力が減ってしまう。早くジャンプしないと瞬く間にオダブツだ。

one point column ワンポイントコラム

仰天の滑らかスクロール

MSX1には、MSX2や2+で追加されたハードウェアスクロール機能がない。MSX1のゲームが当時のファミコンなどと比べて、ちょっとばかりぎこちなかった理由がそれだ。しかし、本ゲームでは見事なアイデアで"滑らかスクロール"を実現。プログラムに興味ある人なら「いったいどうやってるの？」と興味津々ではないだろうか。そこで本誌p.125に作者自らによって、その秘密を公開してもらった。VDPの奥深さを堪能してくれ。

テセウスの命は、気持ちのよい滑らかスクロール。背景の壁が、ドット単位で動いているというのだから驚きだ。

作者から一言

イリーガスの開発も終わったあるとき、渋谷センター街のゲームセンターで見つけたベクタースキャン方式による全方向スクロールのゲーム「Major Havoc」（アタリ）にハマった僕たちは、このベクタースキャン方式なら簡単にできるスムーズスクロールを、お世辞にも強力とは言いがたいMSXで実現してみたくなったのだ。答えはわりとすぐ見つかった。あ～、こうすればいいじゃん。テセウスのネタができあがった瞬間だった。後はクラブのツテで探した女子大生のK嬢にキャラクタデザインを頼んだり、アスキーが用意してくれたエアコン完備、弁当注文し放題という極楽環境で、仲間にも手伝ってもらって完成させたのだった。売れたか売れなかったかと聞かれたら正直期待ハズレだったのかもしれない。でも、こんな楽しい仕事はめったになかった……というのが20年後の感想だ。（滝口彰）

ローレゾモードによる3D迷路脱出ゲーム

イリーガス エピソード4

ILIGKS Epsod Ⅳ

Windows　Pocket PC

1983年 発売：アスキー
作者：三橋正邦、滝口彰

イリーガス(ΙΛΙΓΞ)、それはギリシャ語で"渦巻き"のことをいう。キミは惑星Al-Mazar(アル・マザール)をシミュレートした環境の中で、命がけの脱出に挑戦しなければならない。果たしてそこに渦巻いているものは何なのか!?　というわけで高速3次元迷路脱出ゲームである。頼りになるのはお日様の位置と方向感覚。キミは、夜の闇、地面の穴、巡回ロボットなどの恐怖に打ち勝って出口を目指すのだ。さまざまなアイテムを上手に利用しよう。

ゲーム解説

イリーガスの操作

キー	意味
[↑]キー	前進する
[←][→]キー	左右の方向に向く
前進中に[↓]キー	ジャンプする
[スペース]キー	レーザー発射
[F1]キー	時計のハック
[F2]キー	赤外線ヒュアーのオン／オフ

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

スタミナ付けて延命だ

動きのスムーズな擬似3Dによる迷路脱出ゲームである。のっけから技術的な話になってしまうが、このゲームは、MSXの画面モードの中でほとんど使われずじまいだったSCREEN3(64×48ドットの低解像度モード)で作られているところでMSXファンはまず「ほ～う」と唸っていただきたい(ど～ぞ…ハイおっけー)。

全体的にモザイク調なのはそのためだが、プレイしているうちに気にならなくなってくるから不思議である。

ゲーム内容だが、とにかく迷路を進み、出口を目指せばよい。ただし、あなたのスタミナはAl-Mazarと呼ばれるこの世界の時間でほぼ1日分しかない。スタミナが切れるまでに迷路を脱出するか、途中スタミナの補給をして延命しよう。スタミナは、落ちている水か食糧を拾えば補給できるが、完全に切れたらゲームオーバーになってしまうから、とにかく急げ～。

夜中の友は赤外線ビュアー

Al-Mazarでは、時間が刻一刻と進んでおり、それに応じて太陽も動いている。ゲームの開始時間は常に12:00からである。まず、塀の上にチラチラ見える太陽で方向を把握しよう。太陽が沈んで夜になると辺り一面が真っ暗になってしまう。この中で活動するには赤外線ビュアーが必須だ。どこかに落ちているはずだから拾うべし。赤外線ビュアーのエネルギーが切れると、迷路が表示されなくなってしまう。[F2]キーで作動／停止が切り替えられので、マメに停止してバッテリーを大切に使うといいだろう。バッテリーが切れても、画面下の方向指示器(赤外線ビュアーを作動させたときの向きを、ずっと示してくれるコンパスのような装置)だけは別のリチウム電池で動作しているので、その後もこちらは利用できる。カンだけで進むよりもずっと安全に進めるはずだ。なお、バッテリーを拾えば充電され、再び使えるようになる。

敵は大穴と巡回ロボット

スタミナ切れの他にも、ゲームオーバーになる要素が2つある。1つは、地面に空いた落とし穴。ここに落ちるとあっけなく死んでしまい、ゲームオーバーとなってしまう。その先に進むには、落とし

穴の上をジャンプして渡ればいいのだが、ジャンプの操作はちょっとムズい。具体的には、まず穴の近くまで[↑]キーを押し続けて助走し（助走するスペースがなければ渡ることはできない）、直前で[↓]キーを押すのだ。行き過ぎると穴に落ちてしまうし、手前すぎると渡れるのだろうかと心配になる。ともかく、何もない長い通路で十分練習を積んでから挑んでほしい。

もう1つの要素は、迷路内を移動している巡回型監視用ロボットだ。これに捕まってしまうと、やはりゲームオーバーである。ロボットが接近してくると探知機のヴ〜ンヴ〜ンという音の間隔から、ヴンヴンと短かくなっていく。キミの心臓もどっきどきだ。ともかくこれで距離を把握して、出会ってしまったら逃げるか、レーザーで破壊しよう。巡回ロボットは一定のルートを規則正しく移動しているだけなので、巡回ルートを外れた位置にいれば、ロボットが襲ってくることはない。そこから攻撃すれば安全だ。なお、ロボットは1体だけとは限らないので、そこは注意だ。また、ロボットの巡回経路と同様に、アイテムの落ちている場所は毎回同じだから、迷路のマッピングをして攻略に役立てていただきたい。

この世界では時間が常に経過しているので、じわじわ薄暗くなってきて、最後は漆黒の闇につつまれる。あっ夕焼け…なんて余裕ぶっこいてる場合じゃないぞ！

中央の四角っぽいのがロボット。十字架タイプもいる。近づいてくると探知機音の間隔が短くなってきて、自分の心臓もドキンドキンと高鳴ってしまうのだ。

赤外線ビュアーを作動させると夜もバッチリだが、画面下のバッテリー残量メーターに注意だ。ビュアー作動時の方向を示し続ける方向指示器も活用すること。

アイテム解説

赤外線ビュアー

深夜になると迷路は真っ暗になってしまうが、コレを使えば迷路がはっきり見えるようになるのだ。下の方向指示器も役立つぞ。

バッテリー

赤外線ビュアーを使いすぎるとバッテリー切れを起こす。これは困ったなにも見えない。再度使うには、これが必要ってこと。

食糧

スタミナが増えて活動時間がドーンと延びてくれるのだ。見つけたら必ず拾っておくこと。飢え死にはしたくないもんねー。

水

食糧同様に拾うとスタミナが増え、活動時間が半日以上延びてくれる。乾いたノドをうるおして再出発といこうじゃないか。

黄金

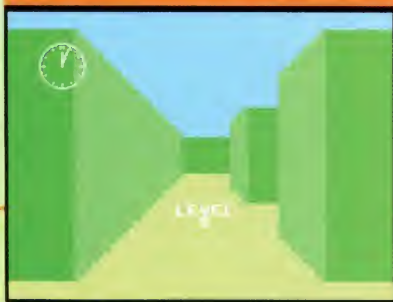
特に効果なしのアイテムだが、お金を拾えば、なんだか嬉しいのは人としての性というものだが、買い物ができなくてはなぁ。

落とし穴

道幅いっぱいのデカい穴。一度落ちるとそこで死んでしまうのだ。キビシイなー。夜は赤外線ビュアーがないと見逃すぞ。

one point column ワンポイントコラム

他のステージから始める裏技

たとえば[5]キーを押しながら[スペース]キーを押せば、いきなりレベル5に行ける。レベルが高いステージをクリアするには地道なマッピングが必要かも。迷路脱出の定石である右手／左手法でも脱出可能だが…。

イリーガスはレベル0〜9の全10ステージが用意されているのだが、実は途中のステージからスタートできる裏技がある。なかなかクリアできないステージは、この機能を使って何度でもチャレンジ！ やり方は簡単。ゲーム開始時に、行きたいレベル数の数字キーを押しながら[スペース]キーを押せばよい。たとえば、[9]キーを押しながらスタートすれば、いきなり最後のステージに挑戦できる。脱出できるまで何度でも挑戦しよう。

作者から一言

MSXの開発用マシンは基盤剥き出しのプロトタイプ機だった。開発マシンはPC-8801＋DUADという構成。最初はバイナリの転送に手がかかっていたのだが、後半はディスケット経由で簡単にテストできるようになって、悪くない環境。資料は、朝日新聞社の「MSX早わかり事典」がとても役立った。作業は、三橋君との分業でやり、完成も比較的早かったと思う。APPLE Ⅱの「エポック」に似せて「オリオン（PC-6001用）」というゲームを作ったことがあった僕は、同じ作者の「ミノタウルス」に憧れがあったのだ。そこで、どうしてもギリシャ語が使いたくなり辞書をあさったところ「ΙΛΙΓΞ」という単語を発見。これがタイトルになったというわけだ。（滝口彰）

隠された宝石を探して船内をくまなく調べろっ！

たわらくん

TAWARA-KUN

Windows　Pocket PC

1984年 発売：アスキー　作者：宮崎暁

昔むかし、嵐で座礁した船があったとさ。その船には、米俵や、味噌樽がたくさん積み込まれており、ダイヤモンドまで隠されていたのだ。"たわらくん"は、ダイヤモンドを探しにその船に入っていくことに。ところが、その船は幽霊船だったのであった…。船底から海水がドバドバ浸入してくる中、果たして"たわらくん"は、宝を見つけて無事甲板まで生還できるのか！

荷物運びの腕がゲームを左右する

ゲームタイトルからすると「俵を運ぶゲーム？」と安直に想像してしまいがちだが、実際には船内からダイヤモンドを探し出すのが目的。浸水が進む船内で、幽霊やサメといった敵キャラを避けつつ、各フロアのどこかにあるダイヤモンドを探さなければならない。

ゲームが開始したら、たわらくんを操作して下へと降りよう。まず障害となるのが、あちこちにある米俵。たわらくんは米俵があると先に進めない。そんなときには[スペース]キーを押す。これでたわらくんは米俵を1つ持ち上げてくれる。あとは、そのまま歩くなり、背後に下ろすなりしよう。米俵を持ったままだと歩く速度が遅くなるぞ。

米俵を持っているときに[スペース]キーを押すと置く。ただし階段の下やハッチの周辺には置けないのだ。また、米俵は2段重ねができる。2段目に重ねるときには、たわらくんを1段目の米俵から半キャラ程度離しておくこと。

敵キャラ退治は米俵とハッチで

船内には、幽霊やサメが始終うろうろしている。しかも、米俵をものともせずに移動するやっかいものだ。そんな敵を退治するには、手にした米俵をぶつければよい。たわらくんを左右に歩かせながら[スペース]キーを押すと米俵を進行方向に投げるから、敵にぶつけよう。ただし、米俵を投げることのできるのは12回だけ。しかも投げるたびに減点されるので、やみくもに投げるのは禁物だ。幽霊はハッチの蓋をぶつけるように閉じても退治できる。なお、レベルが高いステージで幽霊退治をすると左右の壁で跳ね返ってくる。これを計算に入れないと返り討ちにあってしまうぞ。

ゲームが開始すると、時間を追うごとに水かさが増してくる。たわらくんは、水中でも行動できるが、それは画面上に表示されるメーターがゼロになるまで（息の続く限り）だ。息が切れると、たわらくんは悲しいかな溺れてしまうから、その前に水から出て回復を計ろう。

ゲーム解説

たわらくんの操作

キー	意味
[←][→]キー	左右へ移動する
[↑][↓]キー	階段の昇り降り
[スペース]キー	積荷の上げ下ろし
	ハッチの開け閉め
[←][→]キー＋[スペース]キー	手持ちの米俵を投げる

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲームを起動時には、プレイヤー数が選択できる。1人なら[←]キーを押して[1PLAYER]側に、2人なら[→]キーを押して[2PLAYERS]側に矢印を動かそう（1Pと2Pは交互にプレイすることになる）。また[↑][↓]キーでレベルが1～4まで選択できる。レベル2以上を選択すると途中ステージからスタートできるのだ。

レベル2　味噌樽が登場
レベル3　幽霊に俵をぶつけると壁で一度跳ね返る。サメが登場。
レベル4　幽霊に俵をぶつけると壁で二度跳ね返る。サメに二度俵をぶつけると怒り出す

置けない ×

置ける ○

置けない ×

積荷の上げ下ろしのコツ　米俵を手にするときには、米俵にたわらくんをぴったりくっつけて[スペース]キーを押せばよい。しかし、米俵を置くときには、置きたい場所から半キャラ程度離れている必要があるので距離をうまく調節すること。これは2段目に積荷を載せるときや、ハッチを開け閉めするときにも同じだ。位置調節は正確に。

ゲームがスタートしたら、船の中に下りてゆく。置いてある米俵は、たわらくんの障害となるので、持って別の場所に置くなり、投げるなりしよう。

息は水中でも思いのほかもつ。ダイヤモンドを探せるぐらいの時間はあるので慌てることはない。米俵を持ち上げれば、ハイこの通りダイヤ発見！

海水の上昇は、同じ階のハッチを全部塞ぐことで一時的に止めることが可能だ。蓋の上に積荷を載せておけばさらに長時間保てたりもするのだ。

浸水は、各階の床のハッチを閉じると一時的に止めることができる。圧力が高まると自動的に開いてしまうが、それでも時間稼ぎには十分だ。さらに、このハッチの上には1つだけ米俵を置くことができる。こうしておくと、さらに長い間浸水を防げるのだが、幽霊がハッチ上の米俵を見つけると蹴り飛ばしてしまう。あーん、とても憎らしい！

しらみつぶしに宝石を捜せ

ダイヤモンドは米俵の下に隠されていて、米俵を持ち上げれば発見できる。見つけたら、たわらくんを重ねよう。これで拾ったことになり画面の上に表示される。画面の上に表示されるのは、あくまでも拾ったダイヤモンドの数だけで、どの階でダイヤモンドを拾ったのかはわからない。したがって、ダイヤモンドを見つけた階については自分でしっかり記憶すること。覚えておかないと残りのダイヤモンドがどこにあるのかわからなくなってしまうのだ。各フロアにあるダイヤモンドを全部発見しても、安心してはだめ。たわらくんを甲板まで移動して、はじめてステージクリアとなるからだ。最後まで気をゆるめないように。

one point column ワンポイントコラム

味噌樽で高得点をたたき出せ！

ダイヤモンドを全部拾うだけなんて簡単すぎという人は、ハイスコアを狙っていきたい。得点を上げるもっとも簡単な方法は、幽霊退治だ。1匹やっつけると、そのたびに200点もらえる（ただし、米俵を投げるたびに-50点となる）。また、一度に2匹以上の幽霊を退治すれば得点は倍になる。チャンスがあれば積極的に狙っていこう。もっとスコアをのばす方法としては、甲板までの積荷運搬がある。米俵を甲板まで運んでも100点にしかならないが、味噌樽を甲板まで運べば1樽につき1000点も得ることができるのだ。これはオイシイ…かもよ！

味噌樽を4つとも甲板に運んで4000点ゲット。時間もなく、歩行速度も遅くなるので、幽霊の餌食になりやすい。成し遂げるのは大変難しいが、一度はチャレンジしてみて～！

作者から一言

これ、もともとは荷物を全部甲板に運ぶゲームだったんですよ。だけど、作っているうちに「ちょっとゲーム性が弱いねぇ。もう少し何かつけ足そうかー」ということになって、ダイヤモンド探しがメインになっちゃいました（笑）。『たわらくん』というネーミングの由来ですが、これはもう単純に積荷のドット絵が俵ぽかったからです。また、当時は「～くん」てつけるのが流行ってましたので、そのままゲーム名も"たわらくん"でいこうということになりました。一部の営業さんにはウケてましたよ。たわらくんは、音楽を除いて、キャラデザインからプログラムまで僕が一人で作ったんですが、今のゲーム業界では考えられないことですよね。（宮崎暁）

登場キャラ

たわらくん

プレイヤーが操作するキャラ。米俵もしくは味噌樽を1つ持って歩いたり階段を上り下りできる。米俵はエイヤと投げることも可能だ。

幽霊

船の中を始終ウロウロしている敵キャラ。接触すると、たわらくんを1人失う。火の玉状態では無害だが、突然幽霊になることがある。

サメ

接触するとたわらくんを1人失う敵キャラ。レベルが上がると現れ、浸水した階の中を移動する。オシリをカジられないように。

米俵

船内に置かれている積荷。たわらくんは米俵があると、その先に行けないので、手で持ち上げて別の場所に移動するか、投げ飛ばそう。

味噌樽

大きい積荷。米俵同様に移動させることができるが、米俵の倍のスペースがないと置けない。また、味噌樽は重いので投げることができない。

ダイヤモンド

各フロアに1つ隠されているピカピカ光る宝石。米俵を持ち上げること発見できる。4つとも見つけて甲板まで上ればステージクリアだ。

ハッチ

階段の上にある蓋。閉めておくと浸水をしばらくの間防ぐことができる。蓋の上に米俵を載せておくと、より長時間浸水を防ぐことができる。

大阪城の地下迷宮から脱出せよ！

ライズアウト

RISE OUT

Windows　Pocket PC　1983年 発売：アスキー　作者：h.godai

俺はある組織の命令で敵国である日本に乗り込んできた。その命令とは「大阪城に隠されているという忍者の秘密を盗み出してこい」というものであった。なぜいまどき忍者の秘密が必要なのであろうか。まあ、いずれにしろ俺は、命令に従うしかなかったんだが…。俺は、まず古井戸から調べることにした。ところが買ったばかりのロープが切れてしまい、いや、正確にいうと切られてしまい、地下の迷宮に閉じ込められてしまった。そこには正真正銘の忍者が待ち受けていたのである。

ゲーム解説

スコア　白忍者の数　レベル（面数）

ライズアウトの操作

キー	意味
[←][→]キー	左右へ移動する
[↑][↓]キー	はしごの昇り降り
[スペース]キー	サブマシンガンを撃つ
[CTRL]+[R]キー	自殺

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲームを起動したら、一度[スペース]キーを押す。まずプレイヤー数として[1PLAYER]～[4PLAYERS]が選択できる(1P～4Pは順にプレイすることになる)。続けて「SELECT SPEED LEVEL」というメニューが現れる。ここではゲーム全体の速度が、LOW SPEED(低速)、NORMAL SPEED(標準)、HIGH SPEED(高速)から選択できる。一度プレイしてみて、次回から好みで調節しよう。

登場キャラとアイテム

白忍者

プレイヤーが操作するキャラ。設定上は忍者ではないが、身のこなしは赤忍者以上で足がやたらに速かったりもする。追いかけっこには自信あり。

赤忍者

白忍者を捕まえようと追いかけてくる赤いキャラ。足は遅いが、死んでもわいて出てくるという能力があるため、一筋縄ではいかないのだ。

箱

お城が舞台となっているだけに千両箱だと思われる。たくさん拾えば、それだけポイントが得られるが、拾うのがとて～も難しいものもあるのだ。

鍵

面によっては、扉が行く手をはばんでいることがある。そこで必要となるのが扉の鍵。特定の箱の中に隠されていて、拾うと発見できる。

扉

鍵がないと開かない扉と、鍵なしでも開く扉がある。扉は忍者を足止めしておくのにも使えるが、一度開けると再び閉じることはできない。

とにかく昇れ、どんどん昇れ

場所は大阪城。その地下深くに降ろされた白忍者を地上まで導くのがゲームの目的だ。1つのステージは、岩、渡り板、はしご、池などで構成されており、下に白忍者が、上に赤忍者が現れる。プレイヤーは白忍者を直接操作して、追ってくる赤忍者をかわしつつ、ステージ上部に設けられている「次のステージへのはしご」を昇ればクリアとなる。ちなみにステージは全20面存在する。

ステージには、あちこちに箱が落ちている。素直に考えると「これを全部拾わないといけないんじゃないか」いう気がするだろうが、その必要はない。もちろん拾ったほうが得点になるし、扉の鍵が隠されている箱もある。したがって、積極的に拾った方がいいことは確かだが、難易度が高いところにある箱を拾おうとして、失敗続きでゲームオーバーになるならば、素直にあきらめた方が得策かも。

高いところから落ちても平気

ゲームをプレイするにあたり、まず把握しておくべきことは忍者の能力。岩や渡り板の上を走り、はしごの昇り降りができることは当然だが、厳しい訓練のたまものか、どんなに高いところから飛び降りても平気である。うーん頼もしい。ただし、渡り板に対して飛び降りても素通りしてしまったり、池に落ちると死んでしまうというご愛嬌もある。これは、白忍者と赤忍者のどちらでも一緒の特徴だ。

一方、白忍者と赤忍者には違いもある。第一に、白忍者は赤忍者より速く走れる（ちなみに、落下

ステージ1がこれ。自・敵キャラともかなり小さいぞ。最初のうちは独特のルールになかなか慣れないだろうが、ともかく操作をして要領をつかむのだ。

ステージ2は一本道なので、迷うことがない。逆に、赤忍者と必ず出くわすことになるので、やり過ごす対策が必要。最低二度回避しないとゴールできない。

ステージ3は、いきなり難易度がグッと高くなる。ステージをクリアするにはかなりのやり込みが必要だ。それと右下の箱がなかなか拾えない。

"溶ける岩"の変化

	段階	状態
崩壊時	弾1発目	岩の上を渡れる　（通り抜け不可）
	弾2発目	岩の上を渡れる　（通り抜け不可）
	弾3発目	岩の上を渡れない（通り抜け可能）
復活時	1段階目	岩の上を渡れない（通り抜け可能）
	2段階目	岩の上を渡れない（通り抜け可能）
	3段階目	岩の上を渡れない（通り抜け可能）
	4段階目	岩の上を渡れる　（通り抜け不可）

速度も速い)。この足の速さを活かすのが攻略の基本中の基本。赤忍者がもたもたしている間に箱をゲットだ。第二に、白忍者はサブマシンガンを所持している。[スペース]キーを押せば、白忍者の向きに応じて、左右へ撃つことができる。ただし勘違いしてはいけないのだが、これはあくまでも"崩れる岩"を崩すための銃。赤忍者に向けて撃っても効果はないので、使い道をよ〜く考えること。

赤忍者の動きを見切るが勝ち

赤忍者はどんどん白忍者に迫ってくるので、最初は少し焦ってしまうだろうが、実際にはそれほど慌てる必要はない。なぜなら、赤忍者は白忍者の動きに完全シンクロしているうえ、追尾ルールが単純だからだ。具体的には、まず横位置を揃えるように移動してくる。このとき、移動先が崖だろうが池だろうが、何の回避も躊躇もなく、おかまいなしに進む。また、静止している白忍者と横軸がそろったとき、赤忍者の足元にはしごがあれば降りてくる…とまあこれだけだ。逆に言えば、むやみやたらに白忍者を動かさなければ、いずれ赤忍者は静止し、勝手に動いたりはしない。本ゲームには時間制限もないので、この状態でじっくり対策を練ることができるというわけ。

赤忍者の誘導は、ゲームの重要なポイントである。ステージクリアのためには、岩に引っ掛けたり、崖や池に落としたりが必要となるだろう。しかし、こんな単純な動きの赤忍者であっても、いつの間にか追い詰められて、ほとんど自滅状態になることもあるので行動は慎重に。

地形

崩れる岩

白忍者が所持するサブマシンガンで撃つことで、崩れてしまう岩。この岩は再生能力があって、一定時間を経過するとアラ不思議の元通り。

崩れない岩

何をしても崩れない硬い岩。白忍者がサブマシンガンで撃ってもびくともしない。ある意味安心できる地形。しかし、崩せればな〜と思ったりも。

はしご

ここは、昇り降りすることが可能だ。また、はしごからは飛び降りてもオーケー。ただし、うかつに飛び降りて池ポチャしないように。

渡り板

ここは渡ることができる。ただし、ぶら下がったりはできない。また、高所から渡り板の上に飛び降りたときには、そのまま下に素通りしてしまう。

池

迷宮内にある池。ここに落ちると、敵味方関係なく死んでしまうので気をつけよう。ただし、赤忍者が池に落ちた場合には、画面上から復活だ。

one point column ワンポイントコラム

忍法岩崩しの極意をつかめ

ライズアウトは、最初のうちはただのアクションゲームという感じだが、ステージが進むと、どんどんパズル的要素が増してくる。単純に考えただけでは、どうやっても拾えない箱があったり、どうしても先に進めない地形が出てきたりと、悩みどころ満載だ。そんなときに必ず関係してくるのが、先に紹介した"崩れる岩"を崩すのためのサブマシンガンと、そして"崩れた岩"は一定時間が経過すると自動的に元通りになるという法則。この2つを利用することで、必ずや解法は見つかるはずだ。

作者から一言

アクションパズルゲームという新しいジャンルのゲームに挑戦してみました。「ライズアウト」は、当時世界的にヒットしていた「ロードランナー」と同じタイプのゲームです。そのころ、同じようなジャンルのゲームがなかったため、「ロードランナーの真似だ〜」というクレームがありましたね。今から思えばそれほど似ていないのですが…。ちなみに、私自身ロードランナーの大ファンでして、かなり遊んだクチです(当時はAppleⅡで)。実は、この後にロードランナーの家庭用ゲーム機への移植も手掛けました。ロードランナーとライズアウトは不思議な縁で結ばれていたのですね。(h.godai)

2人同時プレイが可能なカードめくりゲーム

ペアーズ

PAIRS

Windows　Pocket PC

1983年 発売：アスキー　作者：h.godai

いつもニコニコのコッペとイート。彼らを操作して、ステージ上に並んでいるカードをめくろう。2枚連続して同種のカードをめくることができれば得点だ。ただし、ステージ上にはお化けのモンちゃんがウロついていて、カードめくりを邪魔してくるぞ。頼りは、カードを記憶する力と素早い操作。これは本当にモタモタしてはいられない。どひ〜い、大混乱だー！

リアルタイム系の神経衰弱

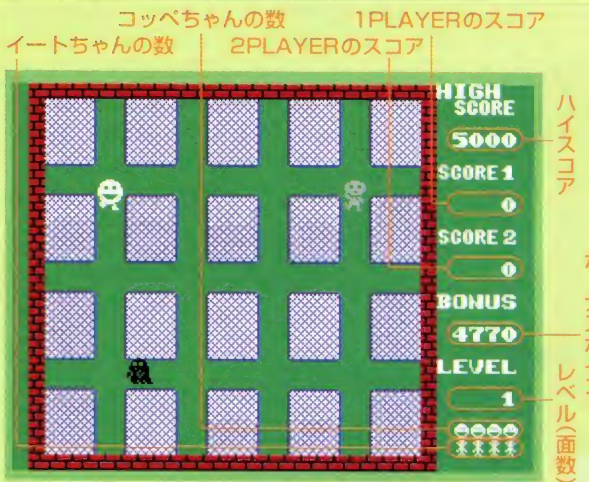
アクションゲームとカードゲームの神経衰弱を足して2で割ったようなゲームがペアーズだ。一見すると、ステージはまるで迷路だが、実は裏返った20枚のカードが並んでいるのだ。このカードをめくって、同種のカードのペアを作り、すべてのカードを裏返すのが、このゲームの目的だ。

カードをめくるには、カードの方向の矢印キーを押しながら［スペース］キーを押せばよい。1枚目を開いたら続けて2枚目を開く。これでペアができれば100点。カードが消え、そこが歩けるようになる。ペアにならなければ最初に開いたカードがふせられる。開いたカードは自分で記憶して、どんどんペアを作っていこう。ゲームが始まると、ボーナスポイントが5000点からどんどん減っていく。早めにステージをクリアすると高いボーナスポイントが得られる。逆にボーナスポイントがゼロになってしまうと、プレイヤーキャラが残っていてもゲームオーバーなので、とにかくスピード最優先で行こう。

お化けが来たらトンズ〜ラだ

カードの絵柄は、1面目がくだもの（ぶどう、レモン、りんご、みかん、メロン、さくらんぼ、バナナ、パイナップル、トマト）、2面目は野菜、3面目は動物となっている。その中でペアになっていない1枚だけのカードが存在するのだが、これについては"MSX坊やカード"とペアになる。"MSX坊やカード"は「どのカードとでもペアになるオールマイティなカード」ではないので注意だ。

本ゲームに緊張感を与えているのが、敵キャラであるお化けのモンちゃん。プレイヤーキャラはモンちゃんに触れただけで1人失ってしまうので、近づいてきたらトンズラだ。初期ステージでは、ほとんどプレイヤーキャラに関係なくウロウロしているが、ステージが進むとモンちゃんの数が増えたり、追いかけてきたり、動く速度が上がってくるのでメチャ大変。

ゲーム解説

ペアーズの操作

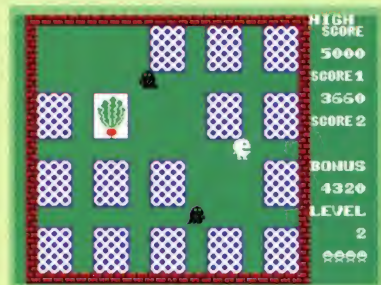
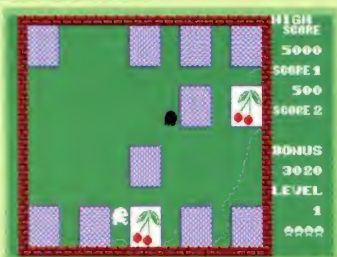
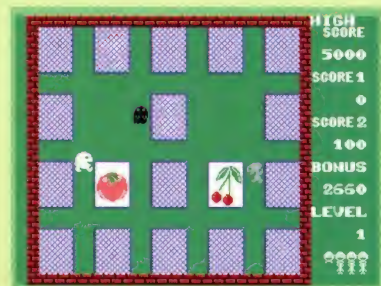
	キー	意味
操作1	[↑][↓][←][→]キー	上下左右へ移動する
	[スペース]キー	カードをめくる
操作2	[2][W][Q][E]キー	上下左右へ移動する
	[TAB]キー	カードをめくる
操作3	ジョイスティックのレバー	上下左右へ移動する
	ジョイスティックのトリガーボタン	カードをめくる

注　どの操作でプレイするのかはゲーム起動時に選択する

起動時の操作　ゲームを起動したら、一度[スペース]キーを少し長めに押す。「HOW MANY PLAYERS 1-2」と表示されたら、プレイヤー数として[1]キーか[2]キーを押す（1なら1人プレイ、2なら2人同時プレイになる）。次に「HIT BUTTON PLAYER 1」と表示される。これは、1P側（コッペちゃん）が何で操作するかの質問だ。矢印＋[スペース]キーで操作するなら[スペース]キーを押す。[2][W][Q][E][TAB]キーで操作するなら[TAB]キーを押す。ジョイスティックで操作するならトリガーボタンを押す。2人プレイの場合には、続けて「HIT BUTTON PLAYER 2」と表示される。これは、2P側（イートちゃん）が何で操作するかの質問だ。ここでは1P側とは異なる操作を選択しよう。

2枚のカードをめくってペアができれば、両カードがステージから消滅する。ペアができなければ、前に開いたカードはふせられてしまうのだ。

表にしたカードを覚えて、どんどんペアを作っていく。カードが消えていけばいくほど、動きやすくなり、お化けから逃げやすくなるのだ。

カードの残りが4枚にまでなった。ここまでくればステージクリアは目前だ。ボーナス得点を得るためにも、のんびりしないで一気に行こう。

先のステージになると、このようにお化けのモンちゃんが2匹に増える。さらに進むと、モンちゃんの速度がアップする。これが滅茶苦茶キツイ！

2人プレイのときには、2人がそれぞれカードをめくってペアを作っていくことができるから、ステージクリアも速攻だ。どんどんペアを作ろう。

作者から一言

MSXが家庭用のパソコンという位置付けだったので、子供から大人まで楽しく遊べるパズルゲームということで考えました。トランプの神経衰弱に似ていますが、順番にめくるのではなく、自分のキャラクタを操作してリアルタイムにめくれるのが特徴です。2人プレイの場合、相手がカードをめくったとき、それと一致するカードをめくると得点が自分のものになるので、そこに駆け引きが生まれます。是非2人で熱くプレイしてみてください。(h.godai)

登場キャラ

コッペ

1プレイヤーが操作するキャラ。丸い身体と、丸くて小さな尻尾が特徴だ。いつも笑顔で元気いっぱいなのが頼もしい。

イート

2プレイヤーが操作するキャラ。ETみたいなデザインで体力的に弱そうな感もするが、能力的にはコッペと同じなのだ。

モンちゃん

黒いお化け。プレイヤーキャラが触れると1人失ってしまう敵キャラだ。面が進むと数が増えたり速度が上がったりするからキツイ！

カード裏

ステージによって異なるものの、同じステージならば、すべてのカードの裏の図柄は同じ。違いを探しても無駄ヨン。

カード表

カードの表には"くだもの"などの絵柄が描かれているが、バナナ食いてぇなぁ、なんて見とれているバヤイじゃないぞ。

MSX坊や

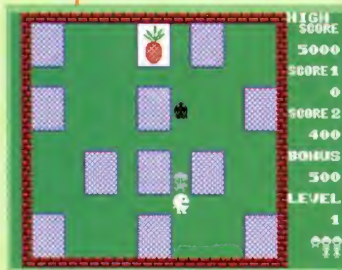
毎回出てくるトラップカード。何かのカードとペアになる。カードが減ってこないと、何とペアになるのか推測が困難。

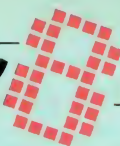
2人プレイでどんどんめくれ！

本ゲームは1人でも遊べるが、2人プレイも可能だ。2人プレイといっても、交代でプレイするのではなく、同じステージ上で2人が同時に遊べるから、とっても楽しいぞ。これは、パーティーなどにはもってこいだ。

2人プレイにするにはゲーム起動時に「2PLAYER」を選択し、次にそれぞれの操作を選択する。ゲームが開始するとステージには、1P側が操作するコッペちゃんと、2P側が操作するイートちゃんが登場（ちなみにキャラ点滅中は無敵だ）。あとは1人プレイのときと同様に、お互いにカードをめくってペアを作っていけばよい。ちなみに2人プレイでは、各自がめくったカードがそれぞれ表になってくれるので、どんどんペアが作れるぞ。速い速い！

one point column　ワンポイントコラム

2人プレイなら意地悪もしてみたい

2人プレイのときには、コッペちゃんとイートちゃんは重なることができない。通路を塞いで、お化けのモンちゃんの餌食にするという意地悪なんかも可能だ。短時間サバイバルで勝敗を決めたいときにはやってみて。

2人プレイでは、互いに協力し合ってステージクリアを目指すという遊び方もできるが、逆に得点を争って競うというのも楽しいのだ。たとえば、コッペちゃんが表にしているカードと同じカードをイートちゃんがめくれば、得点はイートちゃんのものになる。つまり、とにかくペアを作った方の得点になるというわけ。また、ステージ上ではコッペちゃんとイートちゃんを重ねたり、通り抜けたりすることはできない。これを利用すると、相手の逃げ道を通せんぼして、お化けのモンちゃんに襲わせる、なんていうこともできる。あーん、意地悪ぅ！

カードをローテーションさせて並べよう！

ローターズ

ROTORS

Windows　Pocket PC

1984年 発売：アスキー　作者：h.godai

ペアーズに登場したコッペちゃんとイートちゃん、そしてお化けのモンちゃんが再登場だ。今度のゲームはカード合わせ。縦横斜めにカードを揃えると、ドンと得点になっちゃう。カードを移動するためには、カードをポンと蹴ればよい。これでカードがローテーションするのだ。カードをダブルで揃えたり、連鎖消しを狙えばドカンと高得点。これは楽しい。思い切りハマってちょ！

カードを蹴って ローテーション

ペアーズと同じスタッフが作った続編ゲームだ。登場キャラが同じなので、代わり映えしない印象を受けるかもしれないが、実際にはゲーム性が格段に向上していて、いっそう楽しめるのだ。

ゲームが始まると画面に並んでいるカードがシャッフルされる。このでたらめに配置されたカードを移動して、縦横斜めに同種のカードを5枚揃えるのが目的。カードを揃えると、その場所がMSXマークになる。こうして、カード全部をMSXマークにすればステージクリアだ。カードを移動するには、プレイヤーキャラをカードの方向に進めながらカードを蹴る。これで、蹴った方向にカードが順繰り移動する。また、一番遠くにあったカードは手前に戻ってくる。このクルッと回ってくる動きが"ローターズ"（ローテーション）たるゆえんだ。なお、カードを蹴る回数に制限はないので、いろいろな方向から、カードを蹴りまくって揃えるのだ。

1列に並べるには カード選びが重要

ステージにはカードが計25枚並んでいる。カードは1列に揃える必要があるので、最初は5枚存在するマークのカード（ステージ1ならイチゴかチェリー）を選ぶ必要があるのだ。5枚以下のカードでは、どうやっても1列に揃わないぞ。

MSXマークは、どのカードの代わりにもなるオールマイティなカードとして機能する。たとえば、3枚がイチゴの絵柄のカード、残りの2枚がMSXマークという組み合わせでも揃ったことになる。また、1枚が通常の絵柄のカードで、残りの4枚がMSXマークという状況でも同じだ。

ちなみに、本ゲームは2人同時プレイが可能だが、その場合には最後の1枚を揃えたプレイヤーに得点が入ることになっている。したがって、タイミングを計って相手を出し抜いちゃおう。

ゲーム解説

ローターズの操作

	キー	意味
操作1	[↑][↓][←][→]キー	上下左右へ移動する
	[スペース]キー	カードを蹴る
操作2	[2][W][Q][E]キー	上下左右へ移動する
	[TAB]キー	カードを蹴る
操作3	ジョイスティックのレバー	上下左右へ移動する
	ジョイスティックのトリガーボタン	カードを蹴る

注：どの操作でプレイするのかはゲーム起動時に選択する。

起動時の操作　ゲームを起動したら、一度[スペース]キーを少し長めに押す。「HOW MANY PLAYERS 1-2」と表示されたら、プレイヤー数として[1]キーか[2]キーを押す（1なら1人プレイ、2なら2人同時プレイになる）。次に「HIT BUTTON PLAYER 1」と表示される。これは、1P側（コッペちゃん）が何で操作するかの質問だ。矢印＋[スペース]キーで操作するなら[スペース]キーを押す。[2][W][Q][E][TAB]キーで操作するなら[TAB]キーを押す。ジョイスティックで操作するならトリガーボタンを押す。2人プレイの場合には、続けて「HIT BUTTON PLAYER 2」と表示される。これは、2P側（イートちゃん）が何で操作するかの質問だ。ここでは1P側とは異なる操作を選択しよう。

ゲーム開始とともに、フルーツのカードがシャッフルされる。5枚あるのはイチゴとチェリーだけだから、まずこれを並べよう。

イチゴが斜めに揃ったところ。イチゴのカードが全部MSXマークになる。この調子で、残りも全部MSXマークに変えていくのだ。

MSXマークの中からは、ときどきお化けカードが飛び出してくる。うっかりこのカードを蹴ってしまうと1人減ってしまうので要注意。

お化けのカードを3枚並べてしまうと、モンちゃんの速度が上がり、制限時間も短くなってしまうのだ。正に地獄なり。

登場キャラ

コッペちゃん

1プレイヤーが操作するキャラ。カードを蹴ると、カードが1枚ずつ移動し、一番遠くのカードが手前にくるのだ。

イートちゃん

2プレイヤーが操作するキャラ。コッペちゃんとは形がずいぶん異なるものの、能力的にはまったく同じである。

モンちゃん

黒いお化け。ステージ内をウロウロしている敵キャラ。触れただけで、コッペちゃんやイートちゃんが1人減ってしまう。

カード

リンゴ、みかん、バナナ、イチゴなどのマークが描かれている。同種のマークを縦横斜めに揃えると得点になる。

レンガ

通路をふさぐ障害。消したり、移動することはできない。このおかげで、お化けに追い詰められることもしばしば。

MSXマーク

カードを揃えると、その場所がMSXマークになる。そして、以降はオールマイティなカードとして機能する。

お化けカード

ときどき、MSXマークの下から出てくる危険カード。これを蹴ってしまうとプレイヤーキャラはあっけなく昇天だ。

MSXマークの下にもカードがあるぞ

MSXマークは蹴っても動かないが、その下には先ほどのカードがそのまま残されていることになっている。したがって、5枚のカードをローテーションする場合、MSXカードが含まれていても、いなくても、5回蹴らないと一周してこないことを覚えておこう。

MSXマークの下にあるカードは、ときどきお化けカードに変化する。そのため、カードを蹴っていると、MSXマークから突然お化けカードが飛び出すことがある。これを蹴ってしまうとプレイヤーキャラは1人失ってしまうので注意。さらに、お化けカードを縦横斜めに並べてしまうと、モンちゃんの動きが速くなったり、制限時間が短くなったりと、メチャ大変なことになってしまう。これはキツ～イ。

プレイヤーキャラは最初5人いるが、お化けのモンちゃんに捕まるか、お化けカードを蹴ると1人失ってしまう。プレイヤーキャラがゼロになるか、もしくはボーナスポイントがゼロになってしまったらゲームオーバーだ。さあキミは何面までいけるかな。

作者から一言

ペアーズが好評だったため、続編となるローターズを作成しました。こちらはよりパズル性の強いものとなり、連鎖を多くすることで高得点が狙えます。2人プレイで駆け引きが白熱することはペアーズと同じですが、ローターズは1人プレイでも十分楽しめるようになってます。慣れてくると操作一発ですべてのカードを消すこともできますので挑戦してみてください。そういえば、後にPC8801mkII用のゲームとして「ペアーズ」と「ローターズ」が一本のゲームとして発売されました。(h.godai)

one point column ワンポイントコラム

同時揃えと連鎖揃えが醍醐味だ

ローターズで高得点を出すには「同時揃え」もしくは「連鎖揃え」のテクニックがものをいう。「同時揃え」は文字通り、2箇所以上をリーチ状態にしておき、1回の蹴りで同時に揃える技だ。一方の「連鎖揃え」は、カードが揃ったときに表示されるMSXマークを利用して、たて続けに他の列を揃える技。同時揃えや2連鎖程度ならば、技狙いで仕掛けることもそれほど難しくない。しかも、こうした技で得られる得点は、最初の100点に加えて、200点×2、400点×4のように倍増する。ともかくこれは狙わない手はない！

右中央にいるコッペちゃんが前のカードを蹴れば、ウサギに続けて犬とカニまで揃うのだ。これで100点＋200点×2で500点ゲット！

真っ白いゲレンデにシュプールを描け！

トライアルスキー

TRIAL SKI

Windows　Pocket PC　1983年 発売：アスキー　作者：h.godai

目を開けば、そこは冬山。雪に覆われた白銀の世界だ。そして冬といえばスキー。そう、今あなたはゲレンデを果敢に攻める選手なのだ。スタートからゴールまで旗門をしっかり通過して、規定時間内に滑り降りなければならない。スピードとスリルを克服しながらも、あちこちにあるアイスバーンや岩をかわし、タイムロスを最小にとどめることが重要だ。スムーズかつ果敢にチャレンジしよう！

ゲレンデをス～イスイと滑降だ

ウィンタースポーツといえば、最近はスノーボードに押されつつあるとはいえ、やはり王道はスキーである。スキーと言っても、さまざまな楽しみ方があるが、本ゲームは滑降・大回転の要素を取り入れたものとなっている。手前に降りてくるスキー選手を上から操作するというのは、なんとも不思議なものだが、スキー気分は満喫できるぞ。

ゲームの目的は「ともかく早くゴールに到着すること」だ。各コースごとに規定時間が設けられているので、この時間内にゴールまで到達しなければならない。規定時間よりも遅れてしまった場合には再トライとなり、3回トライして失敗するとゲームオーバーとなってしまうぞ。

操作だが、[←][→]を使って向きを変えてやるだけである。これで左右に方向転換ができる。また、同じキーの方向に入れ続けることでエッジが立ち、曲がりやすくなる。この、エッジを立てたり抜いたりする手加減もまた重要だ。

旗門をしっかり通過すべし

コースには旗門が各所に立っていて、その間を通過しなければならない。うまく通過すれば「ぴっ」と音がする。ちなみに、旗門のポールを倒してしまっても、音がすれば通過したことになる。本来のスキー回転競技では、ひとつでも旗門を逃すと即失格なのだが、本ゲームでは1ミスという扱いとなり、1秒のペナルティとして処理される。つまり、いくら20秒でゴールしても4回ミスしていれば、20＋4＝24秒かかったことになってしまうのだ。規定時間が22秒ならばアウトとなり、先のコースには進めない。旗門はしっかり通過するのが鉄則だ。

コース上のあちこちには、乗り上げるとコントロールしにくくなるアイスバーンや、乗り上げると転倒してしまう岩があるので、これらはとにかく避けて通りたい。また、コースアウトして森林にぶつかってしまうと、これまた転倒となってしまう。しかもこの場合、再出発した直後に再び木に当たってしまうという繰り返しに陥ることがあるので、なるべくコースアウトしないように心がけたい。もっとも、これが難しいんだけど。

キーの押し下げ時間は短く

トライアルスキーの攻め方の基本は、とにかく早めの対応をすることだ。このあたりは本物のスキ

ゲーム解説

スコア　経過時間　レベル　コース番号　トライ回数　規定時間

SCORE 1820 LEVEL 1 COURSE 4
TIME 12"65 QUALIFY 24"00 TRY 2

トライアルスキーの操作

キー	意味
[←][→]キー	左右に向きを変える
	押し続けると減速して曲がる

注：ジョイスティックでは、矢印キーがレバーに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

ーと同じである。そのためには、マップを頭に叩き込んでおく必要があるだろう。アイスバーン区域を攻略するときにも、最初からキー操作をしなくても済むような入射角でライン取りすべきだし、たくさんの岩がある場合には、無理をしないで通過できるようなコースから攻めたい。また、より高速に下るためには、キーを押している時間をできるだけ短くするのがコツだ。キーを押し続けるとエッジが立つため、激しく減速してしまうからである。キーの押し下げ時間は短いほどタイムが出る。

見た目は、ややレトロ風味なトライアルスキーだが、ゲーム内容は非常にシビア。真剣にプレイしても数ステージがやっとこである。しかし、攻めどころはきっとあるので、せいぜい諦めずに挑んでほしい。

作者から一言

私はMSXのゲームを多数作成しましたが、「トライアルスキー」は唯一の純粋なスポーツゲームです。当時、私はスキーを始めたばかりの初心者でしたが、従来のスキーゲームの操作性があまりに現実と違うので、操作感に注意を払いながら作りました。マップが広いので、データ圧縮をしています。圧縮するのは今ではあたりまえですが、私が作成したゲームの中では初めてでしたね。(h.godai)

ゲームが開始すると、まずはコースの情報が表示される。コース1の長さは120メートル。規定時間は15秒で、現在のコースレコードは12秒07だ。

スタートゲートをくぐって、いよいよ競技開始。高まる緊張を感じつつ、ゲレンデに飛び出そう。画面が上方向にスクロールしてスキー気分も絶好調!

コース上には、青ポールと赤ポールが対になった旗門が次々と現れる。選手が、この間を通過しなかったときには1ミスとなってしまうぞ。

最後の旗門を通過してゴールゲートに飛び込もう。これでタイムが確定となり、得点が表示される。無事通過となるのか。あ〜ん、ドキドキ。

経過時間は9秒44だが、ミスが5回なのでトータルタイムは14秒44だ。規定時間は15秒00なので本コースは無事クリア。ここで「TRY AGAIN」と表示されたら残念ながら再トライだ。

登場キャラ

選手

プレイヤーが操作するキャラ。大胆かつデリケートに操作してやらないと時間短縮は図れないぞ。

旗門

赤と青の2本のポールで構成されたゲート。選手は、この間をちゃんと通らないとミスとなるのだ。

アイスバーン

ゲレンデ内で雪が凍った箇所。選手がここに乗り上げてしまうと一時的にコントロールしにくくなる

岩

ゲレンデの岩が露出した箇所。選手がここに乗り上げてしまうと転倒してしまってタイムロスとなる。

森林

ゲレンデの左右に広がる森。木にぶつかってしまうと、やはり転倒してしまってタイムロスとなる。

スタート

出発地点にあるゲート。ここをくぐっていよいよ競技が開始だ。緊張しながらも気を引き締めよう。

ゴール

到達地点にあるゲート。ここをくぐった時間が規定時間よりも長いと、同じコースの再トライになってしまう。

one point column ワンポイントコラム

幻(?)のコンストラクションモード

トライアルスキーには、自分でコースがエディットできるコンストラクションモードが備わっている(らしい)。作ったコースはテープに保存するようになっているようである。ところが、本書に掲載するにあたり作者のh.godai氏にたずねたものの、そのあたりのところはキレイサッパリ忘れてしまったという。他の関係者にもアレコレ尋ねてみたのだが全滅。果たしてコンストラクションモードはあるのかないのか、そしてどうすればそのモードに入るのか。MSX最大の謎なのでありました。

さる情報によれば、ステージがある程度進むとやり方が出てくるそうだが、コース10までやってみた限り出てこない。ムズくてこれ以上は進めない。とほ〜!

※コンストラクションモードが存在していたとしても、付属CD-ROMのプログラムはテープの保存に未対応のため、データの保存はできないのであしからず。

ASCII CORPORATION PRESENTS
PAI PANIC
© 1983 Y.KOMATSUDA

1 PLAYER LEVEL1 LEVEL2 LEVEL3 LEVEL4
2 PLAYERS LEVEL1 LEVEL2 LEVEL3 LEVEL4

忙しさ満点のアクション麻雀ゲーム

パイパニック

PAI PANIC

Windows　Pocket PC

1983年 発売：アスキー
作者：小松田裕一・イエローホーン

画面の下段に麻雀の牌が、左から右にズルズルと進んでくるぞ。キミは麻雀小僧のジャン吉くんを操って、手にしているハンマーで牌を叩き落とし、役を作るのだ。ときどきサイコロや点棒が現れて、ジャン吉くんの邪魔をしてくるぞ。ジャン吉くんが、これにぶつかってしまうと、あららハンマーが1つ減ってしまうのだ。さて、どうする？

目の前の牌を逃すな

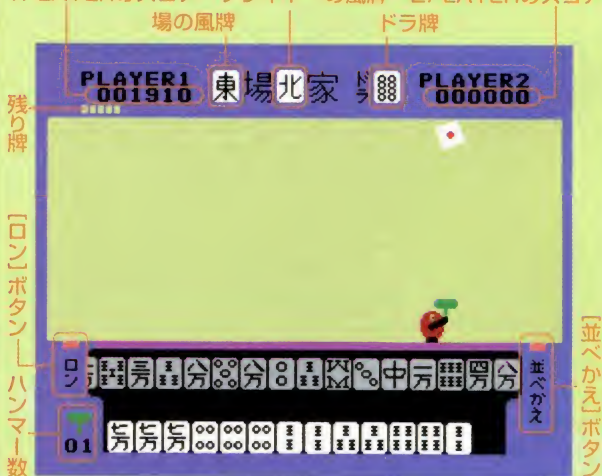
麻雀牌（パイ）がずらっと並んでいることからもわかるように、麻雀をベースにしたアクションゲームだ。しかし、通常の麻雀とはかなり勝手が違うから、その道の達人でもパイパニックには手を焼いちゃう…まさにパニックの嵐！

ゲームを開始すると、中央にプレイヤーが操作する「ジャン吉くん」が現れる。同時に、画面左から右に向かって、数牌（1～10の牌）の萬子（マンズ）、筒子（ピンズ）、索子（ソウズ）、字牌（ジハイ）の東（トン）南（ナン）西（シャー）北（ペイ）白（ハク）発（ハツ）中（チュン）がランダムに流れてくる。必要な牌があったら、ジャン吉くんを近くに移動し[スペース]キーを押す。これで、ハンマーで床を叩き、その下にある牌をツモ（牌を引くこと）することができるのだ。

怒涛のハンマー野郎

ツモをするときには、落とす場所に注意すること。すでに牌がある場所にツモると、もともとあった牌が潰れてしまい、捨て牌（ステハイ）扱いになってしまう。逆に言えば、捨て牌（手牌の交換）したいときには、その上でハンマーを叩いて、下の牌を潰せばよい。また、適当にツモをして手牌（テハイ）が入り乱れてわかりにくくなったときには[並べかえ]ボタンを叩こう。左から、萬子、筒子、索子、字牌の順に並べ替えてくれる。

和了（ホウラー：あがること）したいときには、[ロン]ボタンを叩く。役（ヤク）があればただちに点数計算がされ、役名と得点が表示される。得点は、牌を叩いて落とすと10点、下にあった牌を壊すと10点、あがると1翻につき1000点となっている。和了したものの役がなかったときには「チョンボ」だ。

ゲーム解説

パイパニックの操作

キー	意味
[←][→]キー	左右へ移動する
[↑]キー	ジャンプ
[スペース]キー	ハンマーを打つ

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲーム起動時には、プレイヤー数が選択できる。1人なら[←]キーを押して[1PLAYER]側に、2人なら[→]キーを押して[2PLAYERS]側に矢印を動かそう（1Pと2Pは交互にプレイすることになる）。また[↑][↓]キーでレベルが1～4まで選択できる。レベル2以上を選択すると途中ステージからスタートできるのだ。

レベル1　サイコロが1個登場
レベル2　サイコロが2個登場
レベル3　サイコロが1個、点棒が1本登場
レベル4　サイコロが2個、点棒が1本登場
レベル5　サイコロが2個、点棒が2本登場

お願い 邪魔しないで〜

最初、ジャン吉くんはハンマーを5本持っているが、チョンボをするか、跳ねているサイコロや飛んでくる点棒に触れてしまったときには1つ失ってしまう。サイコロは左右移動で、点棒はジャンプでかわそう。しかし、ハンマーが

ゼロになってしまえばゲームオーバーだ。和了せずに流局（リュウキョク）としてもハンマーは減らないから、役ができなくてもあせることはない。むしろ、役をつくらずに点を地道に稼ぐという手もあるぐらいだ。最後に、パイパニックとしての特別ルールだが、まず親子の区別はない。また、リーチはないので裏ドラもなしだ。ポンもチーもない。そして、門前清（メンゼン）は必ずつくようになっている。というわけで、麻雀を知らない人もパイパニックで、楽しく役などを覚えてみてはいかが？

作者から一言

実は、PCゲームから始まった「ぎゅわんぶらあ自己中心派シリーズ」がこの世に出るキッカケとなったゲームが「パイパニック」です。当時、このゲームが「ゲームアーツ」の宮路社長の目に止まり、片山まさゆき氏の漫画を題材にしたマージャンゲームのプログラマとして白羽の矢を立てていただきました。麻雀のルールも知らなかったくせに気軽に引き受けてしまい、宮路社長には、いろいろな意味でお世話になりました。私がこの業界で飯を食うことを確定付けたタイトルとも言えます。（小松田裕一）

ゲームを開始すると、牌が流れてくる。ジャン吉くんのハンマー使って叩いてツモだ。テコテコ歩くジャン吉くんの動きがユーモラス！

ツモした牌は真下に並ぶから、最初のうちは空いている場所にツモろう。逆に不要な牌があったときには、真上でツモれば捨て牌されるのだ。

レベルが上かると、サイコロは増えるわ、点棒は飛んでくるわと、もう避けるだけでも精一杯。場合によっては避けに徹して点数を稼ごう。

ちゃんと役ができていれは、このように点数計算がされる。役ができてないとチョンボになるから、和了するときにはよ～く確認してからね。

登場キャラ

ジャン吉くん プレイヤーが操作するキャラ。手にしたハンマーで牌をツモることができる。欲しい牌は、どんどん叩いてツモれ。

ハンマー 牌をツモるための道具。ジャン吉にサイコロや点棒がぶつかると1つ失ってしまう。全部なくなるとゲームオーバー。

牌 麻雀の牌。左から右に流れていくので、ハンマーで叩いてツモろう。捨て牌したいときには、ツモ牌で押し潰せ。

並べかえ ツモ牌が入り乱れて手の内がわかりにくくなったとき、理牌（リーパイ）をして並び替えてくれるボタン。

ロン 手牌が揃ったとき和了の宣言をするためのボタン。役があればあがりだが、なければチョンボになっちゃう。

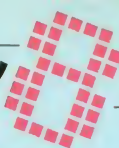
サイコロ 左右からバウンドしてくるお邪魔キャラ。歩いて回避しよう。ぶつかると、タイムロス&ハンマーが1つ減るのだ。

点棒 上から落ちてきて左右に飛んでいくお邪魔キャラ。ジャンプして回避しよう。ぶつかるとハンマーが減ってしまう。

one point column ワンポイントコラム

役と翻数

パイパニックで有効な役を紹介しておこう。翻数が大きくなるほど得点が高くなるのだ。

役	翻数	内容
大三元（ダイサンゲン）	10	白、発、中を3枚ずつ揃える役
四喜和（スーシーホウ）	10	東、南、西、北を3枚ずつ揃える役
字一色（ツウイーソウ）	10	字牌だけの刻子と頭で構成される役
清老頭（チンロウトウ）	10	1と9だけの刻子と頭で構成される役
緑一色（リューイーソウ）	10	索子の23468および発（いわゆる緑色の牌）だけで構成される役
大車輪（ダイシャリン）	10	筒子の2～8を2組揃える役
九連宝燈（チューレンポウトウ）	10	同種の牌で11123456789999と揃える役
国士無双（コクシムソウ）	10	すべての1と9と字牌を1枚ずつ揃える役（頭は2枚）
四暗刻（スーアンコウ）	10	刻子を4組揃える役
清一色（チンイツ）	6	1種類の数牌だけで構成される役
純全帯九（ジュンチャン）	3	数牌の1と9がからんだ順子か刻子で構成される役
二盃口（リャンペイコウ）	3	一盃口を2組作る役
混一色（ホンイツ）	3	字牌と1種類の数牌だけで構成される役
七対子（チートイツ）	2	同じ牌を2枚ずつ揃える役
一気通貫（イッキツウカン）	2	同種の数牌で123、456、789と揃える役
三色同順（サンショクドウジュン）	2	萬子の456、筒子の456、索子の456のように、同じ数の順子を揃える役
三色同刻（サンショクドウコウ）	2	萬子の777、索子の777、筒子の777のように、同じ数の刻子を揃える役
三連刻（サンレンコウ）	2	同種で同じ数の順子を456、456、456のように3組揃える役
三暗刻（サンアンコウ）	2	刻子を3組揃える役
小三元（ショウサンゲン）	2	白3枚、発3枚、中2枚のように大三元から1つ欠けている役
混老頭（ホンロウトウ）	2	1と9と字牌だけで構成される役
全帯ヤオ（チャンタ）	2	数牌の1と9がからんだ順子か刻子と字牌で構成される役
平和（ピンフ）	1	数牌の順子を4組揃える役
断ヤオ九（タンヤオ）	1	字牌と2～8の順子で構成される役
一盃口（イーペイコウ）	1	同種で同じ数の順子を2組揃える役
翻牌（ファンパイ）	1	白、発、中または風牌の刻子があれば1役になる
ドラ	1	ドラがあれば1役になる
門前清（メンゼン）	1	チョンボせずに和了すれば1役になる

浮遊感がたまらない3Dシューティング

レッドゾーン

RED ZONE

Windows　Pocket PC

1985年 発売：アスキー
作者：小松田裕一・イエローホーン

RED ZONE…。かつては緑におおわれた楽園であった。ところがコンピュータに乗っ取られ、いまでは人が近づくことのできない危険地帯となってしまったのだ。そこでキミは「RED ZONE」を人類のもとへ奪還しなければならない。この地は「プロッサー」と呼ばれる監視システムによって侵入者を警戒している。キミの任務は、主力部隊に先立って敵地に侵入し、「プロッサー」を1つ残らず破壊することである。

ゲーム解説

レッドゾーンの操作

キー	意味
[↑][↓][←][→]キー	照準と機体の移動
[スペース]キー	ビーム・キャノン砲を撃つ
[GRAPH]キー	高度の調節
[ESC]キー	一時停止

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンAに、[GRAPH]キーがトリガーボタンBに対応する。

起動時の操作　ゲーム起動時には、[スペース] キーを押す。これでゲームが開始となる。

監視システムを破壊せよ

擬似3Dによる、当時としては迫力のリアルタイム・シューティンググゲーム、それがレッドゾーンだ。あなたはスーパーホバークラフト「コスモパンサー」のパイロットである。ゲームを開始すると、前方からビーム防護壁の「バリー」や敵機「マリポーサー」がつぎつぎと近づいてくる。もちろん敵機は激しくビーム砲で攻撃してくるのだ。それらをかいくぐりつつステージ上の「プロッサー」をすべて破壊するのがキミの使命である。

敵機やプロッサーは自機の左右2門のビーム・キャノン砲によって攻撃できる。照準を合わせてから[スペース]キーを押せば発射。敵もシールドに包まれているので一撃では破壊できないが、何度か当てれば木っ端微塵にできる。ちなみに、ビーム・キャノン砲は連射が利くので、キーを押しっぱなしにしておいてもオッケー。レッドゾーンでは、エリアをしばらく進むと一巡して再び元の地点に戻ってくる。プロッサーを破壊しそこねたときには次回のチャンスにかけよう。

ールド温存を心がけるベシ

破壊すべき残りのプロッサー数は、外視モニターにOBJECTとして刻々と表示されるから、ちらりと確認しておきたい。また、コスモパンサーがプロッサーに近づくと、外視モニターがレーダーに変わり、プロッサーの位置を教えてくれる。これも見逃さないこと。

自機「コスモパンサー」は16レベルのシールドで防御されている。シールドが生きている間は敵のキャノン砲に当たってもダメージはない。しかし、シールドがゼロになってしまうと、次の一撃でゲームオーバーになってしまうから要注意。シールドは、敵ビーム砲に当たったとき、そしてビーム防護壁「バリー」にぶつかったときに減ってしまう。さらに、浮いているコスモパンサーを軟着陸させずに地面に激突させてしまったときにも減ってしまう。急降下すると下降メーターが赤色になって警告してくれるから、緑色に戻した状態で着地させよう。なお、自機の上昇用のキーは[↓]なので最初は少し戸惑うだろうが、戦闘機の操縦桿のように「引けば上昇」だと考えよう。

めの展開が高得点の秘訣！

得点は、敵機を破壊すると与えられるが、このとき同時にクラッシュポイントも増えていく。クラッシュポイントが50ポイント増えるたびにシールドが2増加するので、シールドの回復には敵機への攻撃が条件だ。また、エリアをクリアするとボーナス点が加算される。ボーナス点はエリア数×1万点が基本となっている。ただしこれは、1つのエリアの全プロッサーを1巡目で破壊したときの得点と

ゲームスタートすると早速マリポーサの攻撃だ。落ちろ落ちろー。照準合わせに慣れが必要だが、エリア1はバリーの数も少ないので落ち着いて地上から攻撃だ。

フロッサーを見つけたら破壊せよ。フロッサーのシールドが弱まってくると、このように、外視モニターに表示されるフロッサーの点が赤色から黄色に変化する。

1巡目にエリアのブロッサーをすべて破壊すると、最大のボーナス点が入るのだが、こんなふうにブロッサーが2つ並んでいたら、こりゃ至難の業でござるよ。

自機のビーム・キャノン砲はバリーを通過することを覚えておこう。バリーの裏にブロッサーがある場合には、レーダーで位置だけ確認して撃ちまくるのだ！

ブロッサーをすべて破壊するとステージクリア。このように青色のエリアになり、しばらくの間休息の時間となる。さて、次はどんなエリアが登場するのか!

なっていて、2巡目ならば5千点、3巡目ならば2千5百点と半減していく。高得点を目指すなら早い展開が求められるだろう。

コスモパンサーの操縦だが、地面よりも、やや浮かせておいたほうが、地上への攻撃と空中への攻撃に機敏に対処できそうだ。高度の調節には［GRAPH］キー（Windowsでは左［Alt］キー）を使うのがオススメ。押せば照準の移動と関係なく高度調節ができる。それと、自機と敵機のビーム砲はバリーを通過する。バリーで敵が見えなくても攻撃されるから油断は禁物。常に壁の向こうには敵がいるものとして弾幕をはろう。見えなくても基本は常に攻撃だ！

登場キャラ

ブロッサー

侵入者を監視する地上構築物。中心の赤い部分にビーム・キャノン砲を4発当てると破壊できる。ブロッサーをすべて破壊すればステージクリアだ。

バリー

侵入者を阻むためのビーム防禦壁。自機が当たるとダメージを受ける。自機のビーム・キャノン砲や敵機のビーム砲は、これを素通りするぞ。

マリポーサー

円盤状の敵機。ビーム砲で攻撃をしながら上空へと去っていく。動きが素早いためビーム・キャノン砲を当てるのはなかなか難しい。

ボローサー

26面体の形をした敵機。ぐるぐる回りながら旋回している。赤い目がこちらに向いているときにビーム・キャノン砲を命中させよう。

one point column ワンポイントコラム

好みのステージに直行せよ

簡単にはクリアできないレッドゾーンだが、先のステージを拝みたい人もいることだろう。実はレッドゾーンには裏技があって、これを使うと好きなステージに行けるのだ。その方法とは、まずゲームプレイ中に［ESC］キーを押してゲームを一時停止する。その後で、［BS］キーを押しながら「ステージセレクト一覧表」で示す［1］～［4］キーを押す。さらに、それらのキーを押しつつ［SELECT］キー（Windowsで［PageDown］キー）を押す。これで選択したステージが始まるのだ（失敗するとステージ1になってしまう）。ただし、MSXPLAYerではWindowsマシンの制約によって［1］～［4］キーの同時押しを受けつけてくれないことがあるので、全ステージに自由に行けるとは限らない。行けないステージについては自力でガンバれ！

ステージセレクト一覧表

ステージ	キー
ステージ 2	1
ステージ 3	2
ステージ 4	1 2
ステージ 5	3
ステージ 6	1 3
ステージ 7	2 3
ステージ 8	1 2 3
ステージ 9	4
ステージ 10	1 4
ステージ 11	2 4
ステージ 12	1 2 4
ステージ 13	3 4
ステージ 14	1 3 4
ステージ 15	2 3 4
ステージ 16	1 2 3 4

注：［1］～［4］キーの同時押しが必要なステージには行けない可能性がある。

作者から一言

「STAR WARS」という、当時としては目新しい3D（ワイヤーフレーム描画）表示の大型筐体ゲーム（コックピットに座って遊びました）にハマリ、終電の時間も忘れ、渋谷のゲームファンタジア（ゲームセンター）で朝まで遊んでいたころに作ったのが「レッドゾーン」です。とにかく「STAR WARS」の3Dの迫力は強烈で、プレイをしていると、弾や柱を避けるときに、ついつい体が勝手に動いてしまったものです。「レッドゾーン」では、その迫力と臨場感をMSX1で再現することを最大の目的として作りました。揺れてください。（小松田裕一）

襲いかかる7種類の敵を撃ち落せ！

スコープオン

SCOPE ON

Windows　Pocket PC

1983年 発売：アスキー　作者：宮路武

ガリウス暦2683年に始まった第1次恒星間戦争は地球人類にとって初めての異星人との闘いであった。緒戦での勝利もつかの間、異星人は7種類の戦闘機をもってこれに対抗し、戦闘はこう着状態に陥ってしまった。キミはこの状況を打破すべく開発された新型戦闘機タックに乗り込み、地球軍を勝利へと導かねばならない！

連打連打で飛ぶ敵落とせ

スコープオンは、ひたすら撃ちまくりのオーソドックスなシューティングゲームである。ゲームを開始すると、下方からゆっくりと自機「タック」が出現する。これを操作して前後左右から攻めてくる敵をやっつけるのだ。

タックは、矢印キーで上下左右に移動できる。ただし、飛行範囲は限られていて、左右には画面よりも戦闘機1機分内側、上方向には画面の約半分までしか移動できない。左右や上に追い詰められてヤラれてしまわないように、基本的には中央下で余裕を持って敵の攻撃に応じたいところだ。

光子ビームは［スペース］キーで発射される。一度に2発まで撃てるので、どんどん撃つべし。［スペース］キーを押しっぱなしにすると連射になるが、自分でキーをペチペチと連打したほうが、トータルではより多くの光子ビームが発射できる。なお、タックは5機から始まり、攻撃を受けてしまうと1機失う。そしてタックをすべて失うと、残念ながらゲームオーバーである。

敵の攻撃パターンを見切ろう

敵は7種類いて、それぞれ異なった攻撃パターンで攻めてくる。ゲームを開始して、最初に現れるのは赤いレミークだ。弾は撃ってこないが、ジグザグ飛行をしながら降りてきて、下までくると折り返して戻っていく。ここでは、光子ビームを撃ちつつも、逃げに徹していればヤラれることはないはずだ。次に現れるのは黄色のウォンザック。やはりジグザグ飛行をしてくるのだが、同時に弾を撃ってくる。これをかわして撃ち落とそう。ちなみにウォンザックは、先に進むと点滅するようになる。しかも消えている間は光子ビームが素通りして効かないため、なかなか撃ち落せない厄介な敵である。

その次からはコンビネーション攻撃が始まる。最初はレミークと黄色いボンベットが同時に登場だ。ボンベットは左右下から現れて、左右下へ消えていくので、画面の両端は非常に危険。中央でなんとか対処すべきだろう。この調子で、7種類の敵のさまざまな攻撃かわしながら反撃だ。

ゲーム解説

スコープオンの操作

キー	意味
[↑][↓][←][→]キー	上下左右へ移動
[スペース]キー	光子ビームを撃つ

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンに対応する。

起動時の操作　ゲーム起動時には、[スペース]キーを押す。これでゲームが開始となる。

小さい動きで避けまくれ

敵の点数は200点から始まり、先に進むほど点数が高くなる。同じ種類の敵でも、そのときどきで点数が異なるため「この敵は安いから無視」などと決め付けることは禁物だ。先に進むと、1機撃墜するだけで1000点ぐらいもらえるようになるぞ。

さて、スコープオンのコツだが、まず自機はなるべく左右への動きのみで操作したい。これだけの動きでも、敵を狙ったり、避けたりすることが十分できるからだ。上下にも動くが、本当にイザというときだけにしたほうが動きに余裕が出る。また、このゲームでは敵の撃った弾が必ず下に進むようになっている。横向きに弾を撃ってきたり、やり過ごした敵が背後から弾を撃ってくるといった、不意を突く攻撃はしてこない。ビックリするのは、ボンベットが背後から登場したときぐらいだ（これも中央にいれば問題ない）。したがって、基本的には前方のみに集中すればいいわけだ。このあたりを念頭において高得点を狙ってみよ〜。

最初の敵はレミークだ。弾は撃ってこないものの、不規則なジグザグ飛行で翻弄される。しかも下辺で跳ね返って上へと逃げていくのがトリッキー。

レミークとウォンザックのコンビネーション攻撃。応戦するには、2種類の異なる動きを見切る必要があるのだ。これはキビシい。

3連で攻めてくるのはストライカー。下方にくると、タック側に曲がって体当たりしてくる。ここでは、上に移動してよけよう。

今度はスプライザーとウォンザックのコンビネーション攻撃だ。雨アラレのように降りそそぐ弾をよけまくろう。しかしこの戦いはいつまで続くのだ。

上から降り注ぐ丸い玉がフィレックだ。光子ビームが一切効かないため、とにかく避けるだけ。最低限の動きだけで対処し、あまり頻繁に動かさないのがコツだ。

one point column ワンポイントコラム

MSXPLAYerの調節がキメテだ

スコープオンは、キャラの動きが滑らかで素早い。しかしMSXPLAYerは、起動したときには、より多くのPCでほどよい環境となるように調節されている。そのため、そのままだとフレーム落ちしたような感じで動きがガタガタになってしまうのだ。このゲームの場合には、これではやりにくくてしかたない。撃った光子ビームも、まるでワープしたかのようである。もし、あなたのPCのパフォーマンスに余力があるならば、MSXPLAYerのフレームレートという設定をREALまたはHIGHに切り替えよう。全体の動きがスムーズになる。ただし、パフォーマンスが低いPCだと、ゲームの速度が犠牲になっちゃうので、適度に調節してみて。

スコープオンは、滑らかな動きが命。その動きを見切るには、MSXPLAYerにもガンバってもらわなくちゃ〜！　というわけでフレームレートはREALだ！

登場キャラ

タック
プレイヤーが操作する戦闘機。一撃必殺の光子ビームを撃つことができる。

敵機一覧

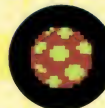
SPRIZER(スプライザー)
攻撃パターン 直線的に突っ込んできてビームをバラバラと発射し、そこで急速に反転。加速しながら上方に去ってゆく。
迎撃対策 大量のビームは、スプライザーが反転したときに発射されるので、その前に撃墜するよう心がけること。

SNOWPER(スノーパー)
攻撃パターン タックに向かって微妙に方向を変えながら体当たり攻撃をしてくる。曲がりっぱなにビームも撃つ。
迎撃対策 中央にいると左右両方からの攻撃を受けてしまうので左右いずれかに移動し、近づいてきた敵だけ攻撃だ。

BOMBET(ボンベット)
攻撃パターン 画面の右下または左下から現れて、グルリと大きな円を描きながら反対側の下隅方向へ飛び去っていく。
迎撃対策 左右下から出てくるので、基本的に中央で迎え撃て。現れた直後に確実に撃墜してしまうのが理想的。

LEMIK(レミーク)
攻撃パターン 速いスピードでジグザグに降下して、下までくると加速し上方に逃げていく。ビームは発射しない。
迎撃対策 動きの予想が難しいので、下の方で光子ビームを連射して弾幕をはり、近寄ってきたら素早く逃げよう。

WONZAK(ウォンザック)
攻撃パターン 点滅しながら降下してくる。消滅中は光子ビームが通過して当たらない。面が進むと消滅時間も長くなる。
迎撃対策 点滅しているタイミングをつかみ、現れた瞬間を狙っていこう。移動速度は遅いので的確に対応すること。

STRIKER(ストライカー)
攻撃パターン 回転しながら縦に3機並んで攻撃してくる。途中まで くると、タックのいる方向に斜めに突っ込んでくる。
迎撃対策 真下で3機ともまとめてしとめよう。真下に潜れなかったときには素直にあきらめて、左右に逃げるべし。

FILEK(フィレック)
攻撃パターン 小型UFOで、上から真っ逆さまに高速体当たり攻撃を仕掛けてくる。光ビームはまったく歯が立たない。
迎撃対策 動きは直線なので怖がることはない。できるだけ小さな左右移動でかわせるように指先を訓練しておくこと。

俺とスコープオン

スコープオンのプログラマは、なんと宮路武氏である。ご存知ない方のために紹介すると、「シルフィード」や「ぎゅわんぶらあ自己中心派」などで有名な株式会社ゲームアーツを兄の宮路洋一氏と設立した人。後にセガサターン向けのRPG「グランディア」のプロデュース、プレステ2版「シルフィード」「ガングリフォン」、ドリキャス版「グランディア2」などを手がけ、現在はケータイ用ゲームの企画プロデュースやコンテンツ事業を主としている株式会社ジー・モードの代表取締役社長である。そんな氏の若かりし頃のプログラムコードが動作しているというのだから、ファンにとっては感慨深いものがあるよね。MSX万歳！

体当たりでやっつけるアクションRPG

イース/イースII/ワンダラーズ フロム イース

［体験版］

Ys/Ys2/Wanderers From Ys ［Trial Edition］

Windows　Pocket PC

1987年の発売以来、ファンを魅了してきたのが「イース」シリーズである。美しい音楽、多様なフィールド、そして誰にでも解けるバランスなど、その後のゲーム界に多大なる影響を与えたことでも有名だ。今回、日本ファルコム株式会社、プロジェクトEGGのご好意により「イース」三部作の体験版（20分の時間制限付き。セーブ／ロード不可）を収録させていただいた。製品版はEGGで購入可能だ。

イース

ゲームは、主人公アドル＝クリスティンが「嵐の結界」を抜けてミネアの町に着いたところから始まる。半年ほど前から呪われた国と呼ばれるようになったこのエステリアの地に待つ冒険とは何か？　まずはむやみに外に出ず、街の人々にくまなく話を聞こう。何をするべきか見えてくるはずだ。

体当たり方式の戦闘

手持ちの金で装備を買ったら[I]キーでインベントリ画面を開き、矢印キーで装備しよう。戦闘はすべて矢印キーによる移動（体当たり）のみで行う。敵にぶつかるとき止まらずに、かつ半分ずらして当たることでダメージを受けずに攻撃できる。これは「半キャラずらし」と呼ばれ、戦闘の基本となる。簡単なようだが、後半では地形が不利になるし敵も賢くなるので、コツをつかむ必要ありだ。

MSX2版イースについて

発売直後から人気が高く、後発のMSX2版はソニーのF1XDを購入した人向けの景品になっていたほど。MSX2版の特徴として、FM音源には未対応の代わりにPSG音源専用の曲がいくつか入っている。「廃坑」「ダームの塔」などはオリジナルのPC-88版と曲そのものが異なる。原体験がMSXな人は迷わずMSX版を選ぼう！

ゲーム解説

イースの記念すべき第一作。オープニングタイトルが非常に美しく、そのセンスにほれぼれ。ゲーム操作がシンプルなのもよかった。

イースの操作

キー	意味
[↑][↓][←][→]キー	上下左右へ移動する
[S]キー	ステータスを見る
[I]キー	持ち物を見る、装備する
[リターン]キー	装備したアイテムを使う 売り買いなどの決定
[ESC]キー	ゲームの一時停止／再開

［ファルコム最新情報］

元気いっぱいの日本ファルコムの新作ゲーム（Windows版）を紹介しておく。まず「VM JAPAN」は今年6月に発売されたシミュレーションRPGだ。1997年に発売された「ヴァンテージ・マスター」のシステムをもとに和風の世界観で構築。8人の幻魔使いから1人を選び、召還した幻魔によって魔晶石を確保しつつ、相手の幻魔使いのヒットポイントをゼロにすれば勝利というシステム。ストーリーの展開と共に使用可能な幻魔や術が増えていくのが楽しい。シナリオモードの他、ネット対戦モードもある。また、2002年の9月には「VM JAPANパワーアップキット～富嶽幻遊記～」が発売され、キャラクタやマップの追加もされた。歴代キャラのリリア（イースII）でも遊べるのがイカすぞ。

もう1つ、12月19日には近年まれに見るハード路線のRPG「ダイナソア～リザレクション～」が発売になる。呪われた傭兵アッシュを始めとして、悩み／苦しみを抱えたキャラクタたちがドラマの中心であり、重厚感たっぷり。ファルコム公式ページには「マッピングが必須」と書かれており、やりごたえの面でも期待できる。そもそもは1990年に発売された「ダイナソア」をエターナル化したもの（単なる移植ではなく、いちから設計して作ること）。ダイナソアはクリアした人が非常に少なかったことで有名だが、そんな人もこちらで再度リベンジしてちょーだい！

シミュレーションRPGの「VM JAPAN」

ハードRPGの「ダイナソア～リザレクション～」

イースII

「イース」は高い完成度を誇ったが、ストーリーにはかなりの謎が残される。1年後に発売されたこの「イースII」はさらに向上したクオリティでその謎を解明できるのだ。まさに夢のようなソフトといえる。オープニングから当時の水準の頂点にあり、思春期のMSXユーザーは誰しもリリアの振り向きにときめいたものである。あの感動を再び！

イースの国へ！

オープニングが終わり「データーディスクを入れて～」と出たらツールバーの「Drive」をデーターディスクに変更して[リターン]キーを押す。ゲームはイース国のランスの村から始まる。前作ではイースと名前がありながらイースは舞台ではなかったのだ。ここから伝説の国「イース」の謎に挑むことになる。基本操作は「イース」とまったく同じ。速度が向上しているので、最初は半キャラずらしの感覚がつかみにくいかもしれないが地形をうまく使って戦おう。

MSX2版イースIIについて

原則として他の機種と変わることはないが、前作に引き続きPSGのみの対応となっている。曲そのものは変わらないが同時発音数が少ないためにMSX2版独自のアレンジがかかっている。メッセージも漢字が少ないために他機種とはビミョーに違う。EGGのPC-88版とMSX2版を比較するのもオツかもしれない。

ゲーム解説

未だに男心わしづかみなイースIIのヒロイン「リリア」。グラフィックもこんなにキレイだ！

イースIIの操作

キー	意味
[↑][↓][←][→]キー	上下左右へ移動する
[S]キー	ステータスを見る
[E](または[Z])キー	武器を見る 装備する
[I](または[X])キー	アイテムを見る、装備する
[リターン]キー	装備したアイテムを使う
	売り買いなどの決定
[ESC]キー	ゲームの一時停止／再開

ワンダラーズ フロム イース

「失われし古代王国」イースの冒険を終えて2年後、アドルは相棒ドギの生まれ故郷フェルガナ地方の異変を知る。たどり着いたレドモントの街は美しかったが、何かが待ち受けている……そんな雰囲気があった。それが、アドルの新しい冒険の始まりだった。

起動時の注意点

オープニングが起動したら、必ず[リターン]キーを押すこと。[スペース]キーを押すとオープニングが飛ばされて、レドモントの街の中からのスタートとなってしまう。ゲームは、2人がレドモントの街に到着したところから始まるから、例によって情報収集から始めよう。なお、ユーザーディスクに対してセーブはできない（エラーは出ないが実際は記録されない）ので注意すること。

アクション性が強調された第三作

前作までのトップビューからサイドビューに視点が変更され、アクション性が強調されている。しかし前作と関係ないストーリーだったため、ファンにとっては賛否が分かれた作品でもある。256色カラーとよく間違えられたが、実際は16色（SCREEN5）だ。全体の速度はゆっくりしているが、アクション部分の操作性は快適。それとMSX2版イースシリーズ全体に言えるのだが、難易度が他機種より低くなっている。スーパーファミコン版がクリアできなかった君も是非どうぞ！

ゲーム解説

これまでのトップビューからサイドビューになって雰囲気も若干変わった作品。キャラもちょっぴり大きくなった印象だ。

ワンダラーズ フロム イースの操作

キー	意味
[←][→]キー	左右へ移動する
[↑]キー	階段を昇る 宝箱を開ける
[↓]キー	階段を下る しゃがむ
[C]キー	剣を振る 会話する
[X]キー	ジャンプ
[I]キー	持ち物を見る 装備する
[S]キー	ステータスを見る
[ESC]キー	ゲームの一時停止／再開
[SHIFT]キー	装備したアイテムを使う

銀河の戦いはかくも無情だ!

アストロモンスターズ

ASTRO MONSTERS

Windows

1995年 © α H7

宇宙歴0095年。太陽系外に勢力を広げる人類の前に「アストロモンスターズ」という生命体が現れた。彼らは地球を目指し侵攻中だという。人類は、対宇宙怪獣用大型万能機動兵器「HERCULES III」の試作機を完成。それはつらく苦しい戦いの幕開けとなった。

極めろシューティング

多彩な敵のフォーメーション、自機の派手なショットとパワーアップシステムなど、どこをとっても硬派なシューティングゲームだ。撃って撃ちまくりながら、敵の攻撃を軽やかに避ける快感に酔いしれよう。最後に立ちはだかるつお〜いボスキャラを倒せばステージクリア。もてる力をすべて出して全7ステージを制覇してほしい。

ゲームを起動するとタイトルメニューが現れる。「OPTION」でレベル(難易度)選択、ジョイスティック選択、音量調節、ショット音のON/OFF設定が行える(CPUとROUNDは効果なし)。また「SOUND TEST」を選ぶとBGMとSEの試聴もできる。肝心のゲームを開始するのであれば、もちろん「GAME START」を選択だ。

強力ショットと硬いボス

敵が襲ってきたら「HERCULES III」のショットで応戦だ。その際、キーやボタンを押しっぱなしにするとセミオート連射になる。しかし、ベタ押しのままだと連射速度が落ちてゆくから適度に連打がお勧め。敵を倒すと、ときどきパワーアップアイテムが登場する。拾えば、攻撃スタイルを変えたり、パワーアップが図れるぞ。一方、自機が敵の攻撃を受けてしまうとライフポイントが減ってゆく(同時にパワーカプセルの効果も下がる)。しかし、その際0.5秒の無敵時間と64発の貫通弾(赤い弾)が使えるので、素早く安全を確保すること。ステージ最後にはボスキャラが登場するが、倒して次のステージに進むとライフポイントが少し増える。ライフポイント0にならないようにガンバれ!

ゲーム解説

アストロモンスターズの操作

キー	意味
[↑][↓][←][→]キー	上下左右へ移動
[M]キー	ショットを撃つ
[スペース]キー	一時停止(PAUSE) 再度押すと解除

注:ジョイスティックでは、矢印キーがレバーに、[M]キーがトリガーボタンに対応する。

アイテム

エネルギーチップ(Eアイテム)

8つとるたび弾の威力が増し、16個で2連弾、48個で3連弾となる。また64個とるたびにボーナス点が入る。ダメージを受けてもパワーダウンはしない。

パワーカプセル(赤/青アイテム)

一定間隔で赤→青に変わる。赤は前方に、青は後方にショットが増える。続けて同色をとると2段階パワーアップ。ただしダメージを受けると1段階パワーダウンだ。

ライフカプセル(Lアイテム)

特定の赤い敵編隊を全滅させると出る。とるとライフポイントが1つだけ回復するようになっている。ライフが満タンのときはボーナス点が入るのだ。

ゲームを始めて間もなくすると、この赤い敵編隊がやってくる。これを全部やっつけると、パワーカプセルが出てくるから、必ず拾うこと。

ステージの最後にはボスキャラが待ち構えているのだ。1面のボスは、2方向弾、3方向弾、そして突撃を仕掛けてくるぞ。こりゃ〜たまらん。

パワーアップアイテムをとることで、自機のショットがどんどん強力になってゆく。見よ、このド派手な3Wayショット。すごい威力だ!

4人同時対戦で熱く迷路をダッシュ！

マジカルラビリンスRemix

Windows　1995-1998年 ©GIGAMIX

父の危篤がきっかけで、莫大な遺産の相続権を求め、4兄弟が同じフィールド(迷路)で激しくバトル。もちろんどの兄弟よりも早くゴールしなければならない。途中、フィールドのあちこちに落ちている6種類のファンキーなアイテムを駆使して、誰もよりも速く勝利をつかもう！

4本の旗を探して迷走だ

1～4人同時プレイが可能なパーティーゲームだ。各プレイヤーは4分割された画面の1つを見ながらゲームを行う。目的は、プレイヤーに割り当てられた色の旗を、迷路の中から4本探すこと。すべての旗を見つけるとゴールがどこかに出現するので、そこに飛び込めば勝利である。

ゲームを起動したら何度か[スペース]キーを押してほしい。メニュー画面が表示されるので「ゲーム開始」を選択。エントリー画面(player entry)に移ったら、カウントが0になる前に各プレイヤーがコントローラを決めて[/][Tab]キーもしくはトリガーボタンを押そう。これでエントリーが確定。早く押した人順にプレイヤー番号(1～4)が割り振られ、各自の色が決定する。最後にステージを選ぶとゲーム開始だ(stage select)。

アイテム使って大逆転

本ゲームで重要なのが6種類のアイテム。それぞれ、自分がいる位置と旗(もしくはゴール)の位置が分かる「マップ」、落とし穴を掘る「スコップ」、自分以外の人を足止めする「クラッカー」、一定時間無敵状態となる「タバスコ」、地面に目印を書く「鉛筆」、壁に穴を開けたり敵や旗を吹っ飛ばす「爆弾」だ。マップ以外のアイテムは使うとなくなるが、宝箱を開くと補充される。「ハズレ」もあるけれど、どんどん拾い集めちゃおう。

アイテムの中には特殊な使い方ができるものもある。たとえば、スコップを2回連続で使うと掘った穴に地雷が埋まる。鉛筆を2回連続で使うとニセの壁(通過可能)が出現する。誰かがクラッカーを投げようとしている間に自分がクラッカーを使えば「投げ返し」も可能だ。こうした要素をうまく活かして熱くバトルだ。でも、あまり危険な行動をすると友達なくすかも～。

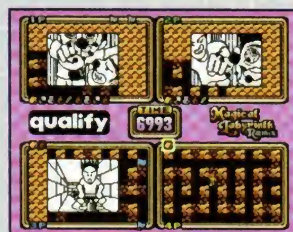
ゲーム解説

マジカルラビリンスRemixの操作

コントローラ	プレイヤー移動	アイテム使用	アイテム選択
キーボード1	矢印キー	[/]	[_]
キーボード2	[E][D][S][F]	[TAB]	[SHIFT]
ジョイパッド1	1Pレバー	1PのAボタン	1PのBボタン
ジョイパッド2	2Pレバー	2PのAボタン	2PのBボタン

MSX実機の絵がリアルなエントリー画面。プレイヤーは名乗りを上げよう。[F5]キーを押せばカウントが強制終了となりゲーム開始だ。

もっとも左のアイテムが「マップ」。この1P画面のように旗の位置が確認できる。マップは他のアイテムと違い何度でも使えるのだ。

クラッカーを投げられてしまうと、他の3人はヘロヘロになり一定時間足止をくわされる。早く復帰したければ左右にガチャガチャ入力せよ！

タバスコを飲んでライバルを吹っ飛ばせ！　やられたプレイヤーは迷路のランダムな位置へと飛ばされ、しばらく気絶してしまうのだ。

のどかな上げモノ？ゲーム

ほいっぷる

Windows　Pocket PC　1992年 ©PASTEL HOPE

ブロックを並べる落ちモノ（上げモノ？）ゲームだ。パズルゲームで有名なPASTEL HOPEの看板的作品である。のどかなCGとほのぼの調BGMが大人気。また、時間制限がないため、じっくり考えてプレイできるのも嬉しい。操作は簡単だけど意外と奥も深いのだ。

押して並べて快感だ

最下段に現れる2つのブロックを押し上げて、1ラインに同種のブロックを並べるのが目的だ。3個以上並ぶと（隣接してなくてもよい）ブロックが消える。並ばなければどんどん積み重なってしまい、天井を突いたらゲームオーバーだ。

ゲームを起動するとメニューが表示される。通常は「START UP」を選んで［スペース］キーを押そう。なお、「DATA CLEAR」はRANKING MODEのランクデータの消去が（確認がないので注意！）、「BGM CLEAR」はBGMの消音が行える。スタート画面で［スペース］キーを押すとゲームモードの選択となる。標準の「NORMAL MODE」は、一定数のブロックを消滅することでステージクリアとなるモードで、全8ステージ存在する。最後までいけなかったときには「CONTINUE MODE」で続きから再プレイ可能だ。

連鎖と高さで高得点

NORMAL MODEでは、一定数のブロックを消すことでステージクリアとなる。画面右上の「ERASE」が、その消すべき数である。ここがゼロになるまで消しまくろう。テクニックの基本は、縦に同じブロックを入れないこと。こうすると、そのブロックどうしは並ぶチャンスがなくなってしまうからだ。また、最下段で揃えられなくても、押し上げによって途中のブロックを揃えることもできるから、全体をよく見てほしい。ブロックを消すと、上のブロックが落ちてくる。これを利用すれば「連鎖消し」が狙えることもお忘れなく。その他、ゲームフィールドの上部で消すほど得点が高くなるとか、魔法のブロックを4つ並べると奇跡が起こり、多量のブロックが一気に消える、といった情報もあるようだぞ。コンプリート目指してがんばれ！

ゲーム解説

ほいっぷるの操作

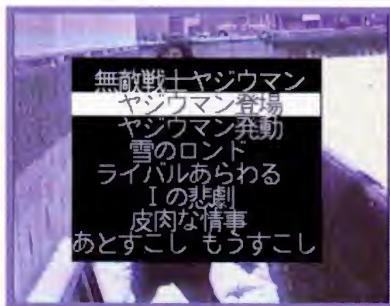
キー	意味
[←][→]キー	ブロックを左右に移動する
[スペース]キー	ブロックを押し上げる
[リターン]キー	ブロックを左右入れ替える

注：ジョイスティックでは、矢印キーがレバーに、[スペース]キーがトリガーボタンAに、[リターン]キーがトリガーボタンBに対応する。

ほいっぷるの基本は、同じ種類のブロックを、縦になるべく入れないことだ。重複しないようにするだけでも、チャンスはぐっと広がるのだ。

このまま押し込むと、最下段で水色ダイヤが揃って消える。すると左の赤い下向き三角が落ちてきて、これまた揃って消える。連鎖って快感。

ステージクリアすると出てくるCG。最初はスープなどが出てくる程度だが、ステージも後半になるとやたらに豪華になっていく。お腹すいた～！

「RANKING MODE」はスコアアタックのためのモードで、ミスするまでずっと遊ぶことができる。高得点をとるとランクインされ、名前が残せるのだ。

変身ヒーローものデジタルコミック

無敵戦士ヤジウマン

Windows

1994-2002年 ©Teamぷらんくとん

いきなり自転車にひかれて死んでしまった太郎少年。しかし彼は、悪の科学者サムソンによって改造され、ヤジウマンとして復活する。サムソンを「パピー」と慕う彼に、果たしてどんな運命が…。MSX2+の自然画モードを活かした怪作を、さあ、キミも今すぐヤジウマッチ！

ヤジウマンがやってきた

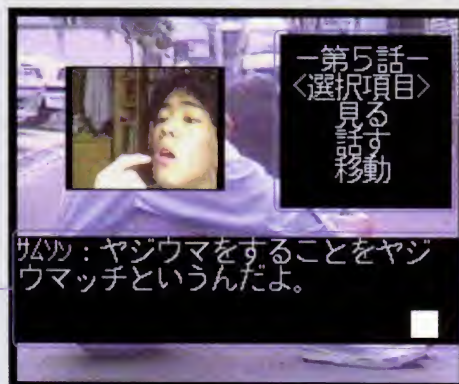
アドベンチャーゲーム（ADV）とも言えるが、"実写デジタルコミック"のほうが近い作品。そもそもは2DDディスク3枚組の大作であり（適宜ディスクを切り替えること）、軽快なノリとテンポ、そして先の読めないストーリーに爆笑のオンパレードである。

ゲームを起動すると「ヤジウマン登場」「ヤジウマン発動」など7本のタイトルが表示される。それぞれがヤジウマンのエピソードになっているので、プレイヤーは上から順番に鑑賞していこう。タイトルを選択すると、ADV風の画面が現れる。ここでは、基本的に［スペース］キーを押していればゲームが進行し、メッセージと実写画像を眺めることができる。ときどき右上の「見る」「話す」「移動」という項目を選択しなければならないシーンもあるが、ややこしいしかけはないので、誰でも最後まで到達できるはずだ。

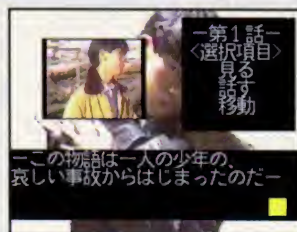
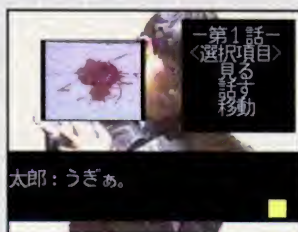
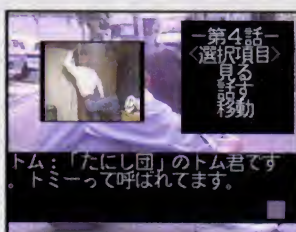
ヤジウマンよどこへ行く

本作の命は、なんといっても体当たりの演技が楽しい画像。MSXのパフォーマンスやディスク容量の制限されている中、パタパタアニメ的な手法とはいえ、これだけの動きをつけているところに脱帽だ。同様に、カメラワークやちょっとしたエフェクトも心得たもの。ちなみに画像は、ビデオデジタイザー「HBI-V1（SONY）」で取り込んだものだとか。また、高速に展開されるアニメーションは、SCREEN 12の裏画面に読み込んでおいた9分割の画像を、左上の領域に必要に応じてコピーをしまくっているらしい。

技術的なことはさておき、ストーリーを最後まで見ると"おまけ"として作者のメッセージが表示される。それによると本作品は92～94年頃、中高生時分に作ったものだそうで、作者曰く「若気のいたり」だそうだが、"青春"って本当にスバラシイですね～！

ゲーム解説

ヤジウマンの操作

キー	意味
［↑］［↓］キー	項目の選択
［スペース］キー	項目の決定、メッセージを進める

注：ジョイスティックでは、矢印キーがレバーに、［スペース］キーがトリガーボタンに対応する。

どうして"ヤジウマン"なのか。たぶんゲームを終えてもよくわからないだろう。だけどそれでいいのだ。だって彼はヤジウマンなのだから！

実写である以上全ての素材を映像化しなくてはならない！ 血しぶきだって同じこと。簡単なようで、やはり情熱のなせる技なのだ。

作り手側がおもしろければそれでいい！ そんなノリがずっと続く。見る側もハイテンションで臨まないとついていけないのだ。

ゲーム中に見ることはできないが、裏画面はこのようになっている。1/9に縮小するのはHBI-V1の機能として存在したもの。

スタート MSXPLAYer

―伝説のマシンがここに復活！―

MSXPLAYerの世界へようこそ！ 全世界で400万台以上が出荷された伝説の8bitマシン「MSX」がWindows、PocketPC2002上にソフトウェアとして復活した。MSXPLAYerとは、「MSX」のハードウェア環境をそのまま他のコンピュータ上で再現させるソフトウェアのこと。従来のエミュレータは権利関係の整理がなされていなかったり、パソコン初心者にはとっつきにくいなどの理由でなかなか注目される機会がなかったが、MSXPLAYerはその点をクリアし、使いやすさを考慮して設計された、MSX史上初の「公式エミュレータ」である。特定のゲームソフトを動かすためのエミュレータではなく、「MSX」のハードウェア環境そのものをWindows上に再現しているため、ゲームを動かすだけでなく、BASICのプログラムを組んで楽しむこともできるのだ。

手持ちのMSXハードウェアが壊れても、ソフトとして蘇った「MSX」は永遠の命を持っているのだから、もう心配はいらない。当時のMSXを知る人は懐かしさを、知らない人は昔のゲームが持っている輝きを、このMSXPLAYerを通じて感じてほしい。

MSX PLAYer MSXPLAYerの全貌

Windows版 MSXPLAYer スキン0ブルー

Windows版 MSXPLAYer スキン1イエロー

MSXPLAYerの使い方はとても簡単だ。以降でくわしく説明するが、インストールや実行の仕方も、通常のWindowsアプリやPocketPC2002アプリとまったく変わりない。まずは、MSXPLAYerの全貌をお見せしよう。

MSXPLAYerには、Windows版、PocketPC2002版ともそれぞれ画面が異なる4つのデザインがある(この画面のことを「スキン」と呼ぶ)。デザインこそ違うが、内容はすべて同じなので、気に入ったスキンを選びインストールして欲しい。もちろん4つのスキンをすべてインストールしても構わない。ゲームごとに違うスキンのMSXPLAYerを起動するなど、遊び方は自在だ。

Windows版MSXPLAYer

収録されるソフトはこれだ！

付属CD-ROMに収録されているソフトは、表に示すとおり。懐かしの市販ゲームタイトルから、アマチュアクリエイターによるプロ顔負けのインディーズゲーム、さらには各種ツール、MSX-BASICまで満載されているぞ。なおWindows版は、インストーラごとに収録プログラムが異なっている。各プログラムを個別にインストール／アンインストールできない点については、注意してほしい（PocketPC2002版は、それぞれ個別にインストールすることができる）。

それぞれのプログラムの詳しい使い方は、該当ページを参照のこと。

Windows対応版収録ソフト

ゴキゲン8bitゲーム (P.10～を参照)

- ボコスカウォーズ
- ザ・キャッスル
- キャッスルエクセレント
- テセウス
- ペアーズ
- ローターズ
- イリーガス
- スコープオン
- たわらくん
- トライアルスキー
- パイパニック
- ライズアウト
- レッドゾーン
- テグザー
- ファイアーホーク
- クインプル
- ファミクルパロディック2

インディーズゲーム (P.48～を参照)

- マジカルラビリンスRemix
- ほいっぷる
- 無敵戦士ヤジウマン
- アストロモンスターズ

BASIC (P.152を参照)

- マウス対応BASIC
- ジョイパッド対応BASIC

グラフサウルス Ver 2.0 (P.102を参照)

- グラフサウルス Ver 2.0

イースシリーズ体験版 (EGG版 P.46を参照)

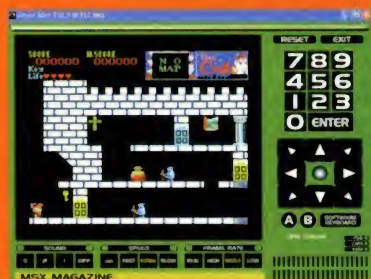
- イース
- イース II
- ワンダラーズ フロム イース

それぞれ20分間プレイ可能な体験版です。製品版は、EGGサイトで購入できます。

PocketPC2002対応版収録ソフト

収録ソフト (P.10～を参照)

- ボコスカウォーズ
- たわらくん
- ほいっぷる
- ペアーズ
- トライアルスキー
- ローターズ
- パイパニック
- イリーガス
- クインプル

PocketPC版MSXPLAYer

Windows版 MSXPLAYer スキン2グリーン

Windows版 MSXPLAYer スキン3ファンシー

PocketPC版 MSXPLAYer スキン0ブルー

PocketPC版 MSXPLAYer スキン2グリーン

PocketPC版 MSXPLAYer スキン1イエロー

PocketPC版 MSXPLAYer スキン3ファンシー

1 INSTALL MSXPLAYER [Windows]

Windows版MSXPLAYerのインストールと実行

MSXPLAYerの全貌がわかったところで、さっそくインストールをしていくことにしよう。まずは、インストールするパソコンのスペックを確認しておこう。

Windows版のインストール方法

付録CD-ROMをドライブに挿入すると、ブラウザが立ち上がる(図1)。ブラウザ画面をスクロールさせて、インストールしたいソフトが含まれるインストーラを選択してクリックする(図2)。ブラウザの設定によっては、[ファイルのダウンロード]ダイアログボックスが表示されることがある(図3)。その場合は、[開く]ボタンをクリックすればインストーラが起動する。あとは指示に従うだけで簡単にインストールが完了する(図4、5)。

インストールの際は、前ページに示した4種類の「スキン」があるので、まずはお好みのスキンを選んでほしい。どのスキンを選んでも、含まれる内容や性能そのものは変わらない。インストーラごとに別々のスキンを選ぶことやすべてのスキンをインストールすることもできるが、ゲームのセーブデータやBASICの仮想フロッピーディスクにセーブしたプログラムは、それぞれのスキンで別々になる点に注意しよう。

なお、「イースシリーズ体験版」については、「EGG」で販売されている製品版の仕様に合わせているため、インストーラは1種類のみとなっている。

アンインストールは、Windowsのコントロールパネルの「アプリケーションの追加と削除」により、該当プログラムを消去できる(図6)。

Windows版の動作環境

Windows版MSXPLAYerの動作環境は、表1のとおり。ひと世代前のパソコンでも十分楽しめる点がうれしい。ノートパソコンなどに、本誌特製シールを貼って「MSX専用マシン」にしてしまう、なんていうのも一興だぞ。

表1 Windows版MSXPLAYerの対応OSと動作環境

OS	Windows98／98SE／Me／2000／XP（2000／XP推奨）
CPU	Celeron500MHz以上（800MHz以上推奨）
サウンドカード	DirectX5以降対応のサウンドカード
ビデオカード	AGP接続の16ビットカラー、800×600ドット以上の表示が可能なもの ※16ビットカラーモード以外で動作させた場合、正常に動作しない場合があります。
その他	ジョイパッド、ジョイスティック対応 フロッピーディスクドライブ（2モード／3モード）対応 ※一部の外付けフロッピーディスクドライブではMSXPLAYerから認識されないことがありますのでご了承ください。

◉インストール

図1 付属CD-ROMを挿入してインストールを開始する。

図2 インストールするソフトを選択する。

注意 MSXPLAYerのインストールの途中で、「セキュリティ警告」ダイアログが表示される場合がある。[はい]をクリックし、インストールを進めてほしい。

Windows版の実行方法

インストールが完了すると、あとは通常のWindowsアプリと起動方法は同じ。スタートボタンから「すべてのプログラム」を選択し、インストールされていることを確認してほしい（図7はすべてのスキンをインストールした例）。メニューはスキン別に階層化されて表示される。

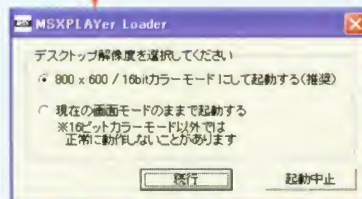
メニューからソフトを選択すると、起動前にダイアログが表示されるので（図8）、通常は推奨の「800×600／16bitカラーにして起動する」を選ぶ。画面モードが自動的に切り替わり、該当のソフトが起動する（図9）。ソフトを終了する場合は、スキンの[EXIT]ボタンを押す。終了してよいかどうかの確認ダイアログが表示される。タイトルバー右の[X]ボタンをクリックすると、確認ダイアログが表示されずそのまま終了してしまうので注意してほしい。いずれの場合も、ソフトを終了すると画面モードは起動時のモードに自動的に復帰するので安心してほしい。

なお、あくまで編集部の調査ではあるが、スキンは「タブレットPC」に搭載された最新のWindowsXPにも対応しているので、ペンで操作することもできる（ゲームそのものはペン操作できない）。

◉MSXPLAYerを起動する

図7 [スタート]ボタンから各ソフトを起動できる。

図8 起動時に画面のモードを変更する。

図9 WindowsでMSXのソフトが起動したぞ！

注意 推奨モードで起動した際、タイトルバー右の最小化ボタンを押したり、[Alt]＋[Tab]キーで切り替えたりすると画面下部のメニューバーが表示されなくなってしまう。そういう状態になったら、[Alt]＋[Tab]キーでアプリケーションを切り替えてほしい。

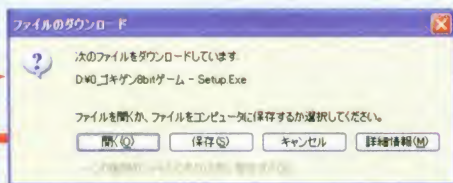



図3 ダイアログが表示された場合は「開く」をクリック。

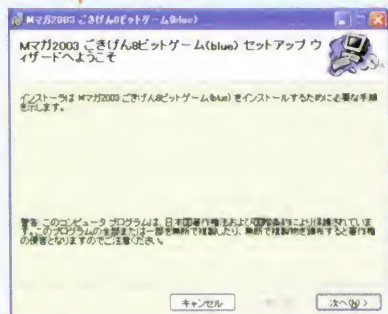



図4 インストールが始まったら手順に従う。

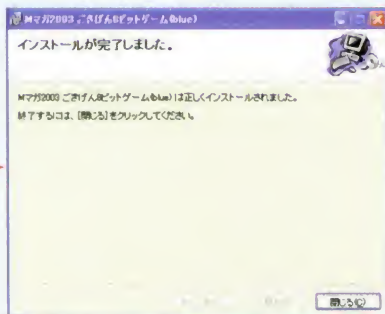



図5 インストールの完了。

◉アンインストール

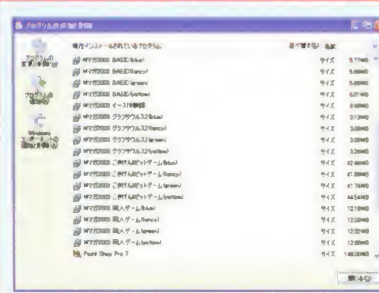



図6 アンインストールする場合は、コントロールパネルの「アプリケーションの追加と削除」から選択できる。

注意 スキンをアンインストールしても、セーブされたゲーム情報や仮想フロッピーディスクにセーブしたデータは残っている。あえて消したい場合は、インストールしたフォルダ内に残っているので、個別に消してほしい。

ゲームソフトで遊んでみよう

MSXPLAYerでは、キーボード、およびスキン上の仮想ジョイパッドでの操作に加え、USB接続のジョイパッドやジョイスティックを使用することができる。複数人でプレイするゲームの場合、最大2本まで接続が可能だ。どちらがプレイヤー1になるかは動作環境によって異なるので、各自で確認してほしい。

それぞれのゲームの遊び方や攻略法などは、「ゴキゲン8bitゲーム」などの該当ページにあるので、そちらを参照のこと。ここでは、スキンの機能と操作について解説しておこう。

DISK CHANGE —ディスクの切り換え—

オリジナルがフロッピーディスク複数枚で供給されていたゲームは、MSXPLAYer内に「仮想フロッピーディスク」を設けて対応している。これらのゲームを遊ぶ場合、途中でディスクを切り換える指示が出た際にはこのスイッチを使ってディスクの切り換えを行う。MSXPLAYerでは、A～Dの計4枚分の仮想フロッピーディスクが組み込まれている。また、「FDD」は実際のフロッピードライブに対応している。詳細はBASIC版の解説を参照のこと。

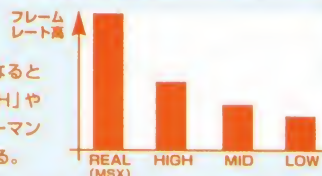
ファイアーホークの画面内の複数ディスク。ディスクを切り替える指示があった場合は、ここをクリックして切り替える。

SOUND —サウンド—

ボリューム調整。3段階の大きさ、およびOFFを指定できる。

ボタン	意味
3	音量大
2	音量中
1	音量小
OFF	音量ゼロ

FRAME RATE —フレームレート—

1秒間に画面を書き換える回数を指定するためのボタン。書き換え回数が多くなると画面が滑らかに動く。動きのスムーズなアクションゲームなどでは、「HIGH」や「REAL」を設定することで、プレイしやすくなる。ただし、その分PCのパフォーマンスが要求されるため、無理な設定をすると全体的な動作速度が遅くなる場合もある。

◉スキンの機能と使い方

FDDランプ

MSXPLAYerがフロッピーディスクドライブにアクセスしていることを示す。

RESET —リセット—

MSXPLAYerをリセットをするためのボタン。リセットするとゲームが再起動する。

EXIT —終了—

MSXPLAYerを終了するためのボタン。クリックすると「終了しますか」とダイアログが表示されるので、終了するなら[はい]を、キャンセルするなら[いいえ]を選択しよう。

SOFTWARE KEYBOARD —ソフトウェアキーボード—

マウスでキー入力を行うための仮想キーボード。切り替えスイッチによってフルキーとテンキーの切り替えができる。キーボードをいちいち使うのが面倒なときや、Windowsの対応キーを忘れてしまったときなどに便利。

SOFTWARE JOYPAD —ソフトウェアジョイパッド—

マウスでジョイパッドやジョイスティック入力を行うための仮想入力装置。一見MSXの「矢印」キー(カーソルキー)のようだが、そちらはソフトウェアキーボードの中に用意されているのでお間違えなきよう。

SPEED —スピード—

文字どおりプログラムの動作速度を示す。実行途中でも自由に切り換えることができるので、ゲーム中の大事な場面では遅く、スキップしてもいいような場面では速く、といった使い分けが可能だ。実機では不可能なエミュレータならではの機能だ。

なお、「SLOW」および「FAST」は、MSXPLAYer内部での動作タイミングを維持したままスピードの制御を行っている。「∞」ボタンを押すと使用するPCのスペックに応じた最大速度でソフトを動作させることが可能だが、MSXPLAYer内部の動作タイミングを維持できず、MSXPLAYer上のソフトが暴走するケースもあるのでゲーム中は押さないこと。

ボタン	意味
SLOW	実機MSXの1/2相当
NORMAL	実機MSXとほぼ同等
FAST	実機MSXの2倍相当
∞	マシンスペックを最大限に引き出す

注意 [NORMAL]ボタン以外は、正常動作の保証が得られないモードとなっている。BASICのDATA文をリードする間だけ使うなど、制約事項を理解の上、利用してほしい。

CAPSランプ

MSXの「CAPS Lock」(大文字固定)モードの状態を示す。Windowsでは[F6]キーでON/OFFの切り替えができる。

かなランプ

MSXの「かなモード」の状態を示す。Windowsでは[F9]キー、もしくは右[Altキー]でON/OFFの切り替えができる。

PLAY BASIC

BASICで遊んでみよう

BASICは2種類のバージョンを収録している。通常バージョンにあたる「ジョイパッド版」と、マウス対応バージョンにあたる「マウス版」だ。通常はジョイパッド版を使用し、マウスが必要なソフトを動作させるときのみマウス版を使用するといった使い分けをしてほしい。なお、BASICのプログラミングについては、p.152の「BASIC講座」などで解説している。

マウス版について

「マウス版」では、Windowsのカーソルと MSXPLAYerのカーソルの切り替えを [F12] キーで行う。[F12] キーを押すことによって、Windows上にあるマウスカーソルが画面から消え、MSXPLAYer上でマウスの動作を検知するモードに切り換えることができる。マウスポインタが消えるので、すぐに区別できるだろう。また当然のことながら、MSXPLAYerのカーソルは、MSX画面の外に移動させることはできない。カーソルが移動できないとパニックにならないように、くれぐれも [F12] キーで切り替えることをお忘れなきよう。なお、「マウス版」には、「注意1」のような制約があるのでくれぐれも注意しよう!

また、グラフィックソフトの「グラフサウルス」は、操作にマウスを使用するのが前提のため、「マウス版」のみが収録されている。「グラフサウルス」の操作は、p.102で紹介しているので、そちらを参照のこと。

仮想フロッピーディスクと実フロッピーディスク

BASIC版では、あらかじめ「A」〜「D」の4枚分のディスクアイコンがセットされている。これらのディスクアイコンのことを「仮想フロッピーディスク」と呼ぶ。「FDD」は、実際のフロッピーディスクのことで、これを「実フロッピーディスク」と呼ぶ。実フロッピーディスクは、PCのフロッピーディスクドライブが割り当てられていて、手持ちのMSX用のフロッピーディスクを読み書きすることができる。フロッピーディスクで供給されていたソフトを遊ぶこともできるし、以前自作したBASICプログラムを読み込んで実行することもできる(図2)。ただし、実フロッピーディスクの使用にあたっては、「注意2」の制約があるので注意してほしい。

これらの5つのアイコンをクリックすることで、実際にMSXPLAYerで使用するディスクを切り替えることができる。なお、仮想フロッピーディスク「A」〜「D」の内容は、表1のとおり。ブランク(空き)ディスクは、「C」および「D」となっているので、ユーザーが各自で作成したプログラムは、そこに保存してほしい。

MSX実機とデータをやり取りする

MSXPLAYerは、MSX2+規格を満たしているため、MSX実機で作成したプログラムのほとんどは、そのままMSXPLAYer上で動作する。また、その逆に、MSXPLAYerで作成したプログラムはMSX実機でも動作する。その際のデータのやり取りには、2DDのフロッピーディスクを利用する。

MSX-DOSを起動する

仮想フロッピーディスク「B」にはMSX-DOSを収録した。ディスクアイコン「B」を選択し、スキン上の [リセット] ボタンを押すことで、MSXPLAYerが再起動しMSX-DOSが動作する(図4)。MSX-DOSの使用方法については、p.174の「MSX-DOS講座」を参照のこと。

なお、MSXPLAYerは、1ドライブ仕様

図1 BASICの起動画面。

- マウス対応のプログラムによっては、マウス入力に正しく反応しない場合がある
- ジョイパッドおよびソフトウェアジョイパッドを使用することができない

表1　仮想フロッピーディスクの内容

仮想フロッピーディスク	収録されているプログラム
A	本書で掲載されているBASICプログラム
B	MSX-DOS1
C	ブランク（空き）
D	ブランク（空き）

図3　仮想フロッピーディスク「A」を選択し、FILESコマンドを実行した結果。

図4　仮想フロッピーディスク「B」を選択し、リセットボタンによりMSX-DOSが起動する。

図2　仮想FDDの仕組み。A～D、FDDドライブはMSXPLAYer上でいつでも切り替えられる。

- 一部の外付けフロッピーディスクドライブは、MSXPLAYerから認識されないことがある。
- 実フロッピーディスクは、原則としてMSX実機でフォーマットされたディスクか、Windows（またはMS-DOS）において2DDフォーマット形式でフォーマットされたディスクに限られる（2HDフォーマット形式には未対応）

なので2つのドライブを同時に使用できない点に注意したい（p.148の「MSXPLAYer Tips集」を参照）。

キーボードの操作と文字の入力

MSXのキーボードには、いまのPCには存在しないキーがある。そのため、以下の対応表にしたがって読み替えて操作してもらいたい。MSXのかな／記号を入力するには、表2にあるように、PCの［F6］、［F9］キーを押す。PCの［CAPS］と［かな］キーを押しても機能しないので、特に注意しよう。

なお、[CAPS]（[F6]）キーと[かな]（[F9]）キーは、押すたびにスキン上のそれぞれのランプがON／OFFするのでどちらの状態にいるかはすぐわかる。

文字の入力方法をまとめると表3のようになる。なお、漢字の入力方法に関しては、「MSXPLAYer Tips集」にあるのでご覧いただきたい。

表2　キーボード対応表

MSX	PC
[CAPS]	[F6]
[かな]	[F9]または右[Alt]
[GRAPH]	左[Alt]
[STOP]	[PageUp]
[SELECT]	[PageDown]
[RETURN]	[Enter]

表3　文字の入力方法

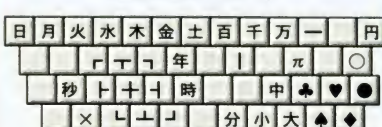
入力文字	操作方法
ひらがな	「かな」ランプが点灯した状態で該当するキーを押す
カタカナ	「CAPS」および「かな」ランプが点灯した状態で該当するキーを押す
記号	[GRAPH]（[Alt]キー）と同時に該当するキーを押す

【記号対応表】

手持ちのディスク版ゲームを遊ぶ

BASIC版では、フロッピーディスクで供給されていたソフトをそのまま遊ぶことができる。以下の手順を参照のこと。

❶BASIC版のMSXPLAYerを起動する。
❷ディスクドライブのスイッチを「FDD」に設定する。
❸フロッピードライブにディスクを挿入する。Windowsがフロッピーディスクの中身を書き換えてしまう場合があるので、ディスクは必ず書き込み禁止状態（タブを開けた状態）にしておく。最悪の場合、MSX実機でも起動しなくなる。
❹MSXPLAYer上のリセットスイッチを押す。

ソフトによっては不正コピー防止のため特殊なプロテクトがかけられており、動作しないものがある。また、フロッピードライブの機種によっては、Windowsの仕様上、MSXPLAYerから認識できない場合があるのでご容赦願いたい。より詳しくは、p.148「MSXPLAYer Tips集」も参照のこと。

PocketPC版MSXPLAYerで遊ぶ

Windows版のMSXPLAYerは、画面も広いし快適でサクサク動くがゆえに、いにしえのMSX実機の哀愁が感じられないという方もいるかもしれない。そんなあなたにお勧めなのが、PocketPC版MSXPLAYerだ。画面もぎゅっと圧縮され、ついにMSXもこんな小さなマシンで動くようになったのかと誰に見せても感動すること請け合い。しかも、ビジネスマシンとしてのPocketPCが、あっという間にゲームマシンに早変わりするのもカッコイイ。

ただし、PocketPC版特有の制限事項があるので、お友達に自慢するのは、以降の解説をじっくり読んでからにしよう。

PocketPC2002版の動作確認機種

MSXアソシエーションで動作確認テストを行い、正常動作を確認したPocketPCを示す。基本的にPocketPC2002規格を満たしている機種は問題なく動作するようだ。

メーカー	機種名
コンパック（HP）	iPAQシリーズ*
東芝	GENIO e550Gシリーズ*
カシオ	CASSIOPEA E-2000
富士通	Pocket LOOX
HP	Jornada568
NTTドコモ	musea
NEC	PocketGear

＊PocketPC2002へのアップグレードしたものを除く

PocketPC2002版の動作確認機種

PocketPCは、機種によりスピードやボタンなどのハードウェアが異なるために、すべての機種で同じような操作性を保証できない点に注意してほしい。MSXアソシエーションでは、上の表の機種で動作確認テストを行い、正常に動作することを確認している。また、PocketPC2002規格を満たしている現行機種については問題なく動作するようだ。

本書では、なるべく多くのPocketPC 2002機で楽しめるゲームのみを付属CD-ROMに収録しているが、PocketPCはゲーム専用に設計されていないため、機種によっては操作性が若干落ちるのは否めない。PocketPCをこれから購入する人は、できるだけ最新機種で、かつ8方向のカーソル入力に対応した機種を推奨する（古い機種やRAMの少ない機種は、スムーズに動作しない可能性がある）。なお、次ページにあるように、機種の問題により固有の現象が発生する場合もあるので注意しよう。

編集部でのお勧め機種は、「HP Jornada 568」「HP iPAQ H3900」「GENIO e550G」の3機種。これらはテストの結果、かなり満足のいく操作性が得られたことを報告しておく。また、編集部で試したところ、残念ながらシグマリオンなどの「Handheld PC 2000」では動作しなかった。

PocketPC2002版のインストール方法

PocketPC2002版のインストールには、パソコン（デスクトップまたはノート）が必要となる。PocketPC2002とパソコンをUSBケーブルで接続し、「ActiveSync3.5」（マイクロソフト社のサイトからダウンロードが可能）を起動して接続された状態でインストールを行う。PocketPCをパソコンと

◉インストール

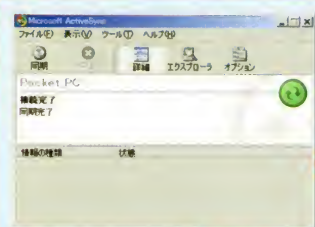
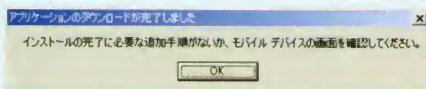



図1　インストールする場合は、まずActiveSyncで同期する。

図2　セットアッププログラムを実行し、ダイアログに従ってインストール。

図3　インストールが完了した。

◉アンインストール

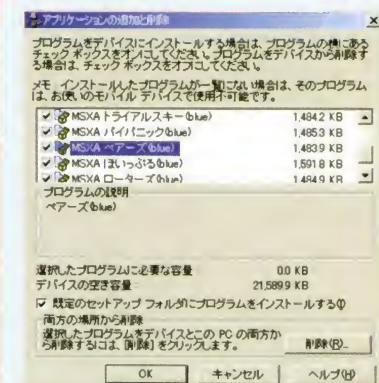



図4　アンインストールもActiveSyncから。

図5　PocketPCにインストールされたプログラム。

図6　PocketPCでMSXソフトが動いた！

接続して使っているユーザーならば、特に問題なくインストールできるはずだ。

付属CD-ROMから直接PocketPC版をインストールすることはできないので、まずインストールプログラムをハードディスクの適当なフォルダにコピーする。自動起動したHTMLファイルでは、該当プログラムをクリックするとインストールプログラムのフォルダが表示されるので、ハードディスクの適当な場所にコピーしてほしい。該当プログラムは、CD-ROM内の「PocketPC2002」フォルダにすべて含まれており、そのフォルダをハードディスクにコピーすれば、一度にすべてのインストールプログラムがコピーできる。

PocketPC版にも4つのスキンが用意されているが、こちらもWindows版と同様に、どのスキンを選んでも、含まれる内容や性能そのものは変わらない。また、スキン別にゲームデータなどが保存される点も同じだ。

続いて、PocketPCとパソコンを接続し、ActiveSyncで「接続完了」状態にしておく(図1)。あとはインストールしたいプログラムを選び、インストールプログラムをダブルクリックして起動する。インストール画面のダイアログが表示されるので、その指示に従えばOK。最後の画面が表示され(図2)、[はい]ボタンでPocketPCへのインストールが開始される。終了画面(図3)が表示されたらインストールは完了だ。

アンインストールは、ActiveSyncの[ツール]－[アプリケーションの追加と削除]メニューでダイアログボックスを表示し、該当プログラムを選択して[削除]ボタンを押すだけだ(図4)。

PocketPC2002版の実行方法

PocketPCにインストールされたプログラムは、[スタート]－[プログラム]メニューで表示される「MSXPLAYer」フォルダにすべて置かれている(図5)。アイコンをタップすればプログラムが起動する(図6)。

ゲームの遊び方

PocketPC版のスキンの機能は、Windows版のサブセットとなっている。PocketPCでは通常、方向キーに相当するボタンと4つの独立したボタンが付いている。方向キーは、そのままジョイパッドの代用として使用することが可能だ。ただし、斜めキーはサポートされていない機種もあり、その場合はスキンの斜めボタンをタップするしかない。8方向の入力ができることが望ましいが、「ボコスカウォーズ」「トライアルスキー」「パイパニック」「イリーガス」は4方向のボタンでも遊べるので、ご安心を。

A、Bトリガにあたるボタンは、右から2つ目のボタンが「A」、一番右のボタンが「B」に割り当てられる。ボタンの名称は機種によって異なり、これに該当しない機種もあるかもしれないので、各自で確認してほしい。

PocketPC版の制限事項

残念ながら、市場に出回っているすべてのPocketPC2002機による詳細なチェックはできていない。以下は、MSXアソシエーション、および編集部でわかっている代表的な機種の情報である点をご了承いただきたい。なお同様の問題は、別の機種でも発生する可能性もある。

PocketPCのボタンは、そもそもゲームの激しい操作向けに作られていない。内蔵されているアプリによっては、長押しやチョイ押し、連打などを行った場合、あらかじめボタンに割り当てられていたアプリが立ち上がってしまう場合もある。また、一部の機種では、ゲームの起動時に音量が最大で立ち上がってしまう。起動直後にボリュームを下げるか、ヘッドフォンを接続しておくなどの対応をしてほしい。くれぐれも予行練習なく、いきなり電車のなかでゲームを起動しないようにしよう。白い目で見られても、編集部では責任を負えないので念のため。

表2　代表的な機種固有の現象と回避方法

機種	不具合と回避方法
HP　Jornada568	ボタンの長押し（2秒程度）であらかじめ割り当てられたアプリが起動してしまう。なお、設定でこの機能は解除できる
東芝　Genio e550G	Aボタンを長押ししつつ、上下キーを素早く押すと、スタートメニューが立ち上がる。回避不能
コンパック(HP)　iPAQ H3600	音の消去を設定していても、ゲームの起動時に音量が最大になってしまう。回避不能

MSX復活のシナリオ！

1チップ化で蘇る次世代MSXの全貌

1983年、統一規格パソコンMSXを世に送り出した西和彦氏。20年の歳月を経て復活した、ソフトウェアMSX、すなわちMSXPLAYerも、仕掛け人はやっぱり西さん。MSXのことは西さんに聞け、とばかりに直撃インタビューをしたが、そこで我々は衝撃の事実を入手した！ MSXPLAYerは、MSXの終着点などではなく、復活の単なる始まりだったのだ。MSX復活のシナリオとは？ 1チップMSXとは？ 西さんの壮大な計画の全貌が、今、明らかになる。

MSXをやってよかった

――MSXの誕生が1983年ですから、2003年でちょうど20周年になります。MSXにとって、20年間とはどういうものだったのでしょうか。

西　20年というのは赤ちゃんが成人になり、12歳の小学6年生が大学を卒業するという、人生的な長さです。今コンピュータ関係の仕事をしている日本人の中で、MSXを使ったことがあるとか、遊んだことがあるという人は、実はすごく多いんです。ファミコンを使って面白かったという子供たちはたくさんいたけれど、それはそれだけ。MSXの場合は、使っていて本当によかったと、コンピュータのことがよくわかったという子供たちがたくさんいます。初めて触れたコンピュータがMSXで、もし、MSXがなかったら自分はコンピュータの仕事をしていなかったという話を今でもよく耳にします。そういう意味で我々は、MSXをやってよかったのではないかと思います。当時、NECのPC-6001もPC-8001もPC-8801も一時代を作った機械ではあったけれど、それらに無いMSXの良さというか、魅力は何だったのかと考えてみると、それはMSXがオープンシステムだったということだと思います。ユーザーが回路もわかっていて、OSもわかっていて、BIOSもわかっていて、こういうふうにしたらどうなるかということがわかっていた。ハードウェア的にオープンで、ソフトウェア的にオープンだったということではないでしょうか。ちょうど81年というのはIBMが16ビットのコンピュータを発売した時でもあるし、8ビットは成熟した時期だったと思うんです。そういう意味でMSXは最後の8ビットシステムではなかったかと思っています。

次のターゲットは携帯電話

――さっそくですが、MSXPLAYerについてお聞かせください。ハードウェアだったMSXがソフトウェアになりましたね。

西　MSXのハードウェアは、Z80のCPUがあって、ビデオプロセッサがあって、メモリがあってという発想でした。ビデオプロセッサのオリジナルモデルとなったLSIに、テキサスインスツルメンツ社の9918というものがありました。MSX（1）には9918を使っていたのですがその後9938という、より高性能なチップをアスキーとヤマハで一緒に開発しました。これらのチップの一番大きな特徴はビデオメモリがCPUの空間から切り離されていたことです。ビデオプロセッサにコマンドを送って円を描くという感じでした。値をレジスターにロードして、IOポートに書き込むという感じでコマンドを送って実行。この方式があとあとグラフィックスに特化したマルチメディアのインストラクションセット、つまり、インテルのMMXとかAMDの3DNow!のようなインストラクションに展開していったと思うんです。グラフィックスプロセッサに対してコマンドを送るという発想の原点というのは、MSXのビデオプロセッサだったと思うんです。今やCPUのクロックが上がり、ビデオの能力も向上した。10年以上の期間を経て、やっとMSXのソフトウェアエミュレータがパソコンの上で動くようになった。こうしたハードウェアの進歩が今回のMSX PLAYerを可能にしてくれたといえます。

――MSXの発想が、PCの性能向上にもつながり、その結果MSX PLAYerが誕生したということでしょうか。

西　MSXPLAYerの誕生については、ハードウェアの進歩だけでは

なくて、もう1つ、インターネットの普及という要素もあります。MSXが生まれた時には、インターネットについて、ほとんど誰も夢にも思っていなかったし、パソコン通信も可能性がわからなかった。しかし、現在のように高速なインターネットが当たり前になると、もはやROMカートリッジでソフトウェアを流通しなくても、ネットワークの上で流通させることができる。この2つによって、今までのMSXとは発想の違った新しいプラットフォームができてきたのではないかと思います。

――すでに、MSXのソフトウェアがMSXPLAYerを使ってオンライン販売されていますね。MSXPLAYerは今後どのように使われていきますか？

西　MSXPLAYerはWindows版とPocket PC版がリリースされています。Linux版やMac版も今、作っていますから、何でも、何処ででも動くようになります。その次は、携帯電話のディスプレイが340×240のクォーターVGAになった時点で、携帯電話でもMSXのゲームが動くようにしたい。そうすると自分が子供の時に遊んだMSXのソフトウェアが携帯電話の中に入ってしまう。そういう計画になっています。とりあえず第一弾としてパソコンで動くMSXPLAYerをやりますが、次は携帯電話にMSXPLAYerが乗るだろうという見通しです。それで、まず自分のパソコンでゲームをする。その次は携帯電話でやる。別にゲーム機じゃないんだけど、ゲームができますよってことで。あとはMSXPLAYerにMPEG4とAACのデコーダをどう載せるかということでしょう。これからネットワークの上の映像配信はMPEG4とAACじゃないかと思うんです。

――携帯電話の上でMSXPLAYerが動いてビデオも見られる。

日本のCPU野郎はどこへ行ったのか？

――まずは、ゲームとビデオ再生ですね。それ以降はどうでしょう？

西　MSXが目指してきたオープンアーキテクチャなハードウェア、オープンアーキテクチャなソフトウェアを通して、コンピュータの初歩を学ぶという世界があると思います。今、コンピュータが好きな人、勉強したい人がどんなことをやっているかといったら、マザーボードを買ってきて、ネジで留めて、インテルのCPUを挿し込んで、それでパソコンができました。私はコンピュータ作ってますとか言っていますが、おいおい本当かい？って話です（笑）。インテルのCPUの中身は、はるかに難しくて、普通に考えたら絶対わからないようになっています。多層のブランチプレディクションとかアウトオブオーダー実行とか。普通のマイクロチップという感覚ではなくて、ミニコンピュータとかスーパーコンピュータとかがシリコンの上で動いている。Pentium4にしてもAthronにしてもそうです。それでどうなるか。みんなハードウェアレベルのことを飛び越してプログラム言語レベルでプログラムを書くようになってるわけ。アセンブラでは書かない。だから、ロジックの世界はあるんだけど、レジスタモデルが頭の中に入っていて、信号がCPUからバスを通ってメモリにいくという、そういうイメージは持てなくなってきている。これが問題です。

――ハードウェアの知識がない。

西　僕の感じでは、どんどんどんどんハードウェアのエンジニアが減ってきている。本当はハードウェアって面白いのに……。実際にハンダ付けするというのが重要なわけではなくて、プログラム言語のレベルでプログラムを考えるというのを、ハードウェアのレジスタレベルにまで引き下げて、そこでロジカルにコンピュータを考えるということが重要なのです。これがコンピュータの教育に一番必要なことじゃないでしょうか。それがない限り、絶対にハードウェアのデザインとソフトウェアは1つにつながらないわけなんです。これは、理科系と文科系みたいなところがあって、文科系の人は理科系には行けないんですが、理科系の人は理科系を抜けて文科系に行ける。それと同じで、ソフトウェアの人はハードウェアに行けないんです。ハードウェアというのは3次元のソフトウェアでもあるのです。ソフトウェアというのはシーケンシャルなロジックだから、ある意味で1次元だとも言えるわけ。そんなものだから結局は、高級言語のロジック展開だけでなくて、レジスタモデルの理解が必要だと思うのです。

――MSXPLAYerがハードウェア技術者を育てるきっかけになる？

西　そうです。日本のCPU野郎はどこへ行ったのか？　最初のCPU、4004、8008、8080、Z80は嶋正利さんが作った。彼は日本に帰ってきて、アスキーと三井物産がお金を出して「VMテクノロジー」という会社を作り、486互換CPUまで作ったんです。これは富士通が生産して、日本のワープロ専用機のシェアの9割を取った。どこのワープロもVMのチップを使っていたんです。その代わり、山ほどインテルに意地悪されて……。インテルはIBMのパソコンと共に86、88を作って286、386、486を作ってPentiumを作ってPentium4を作って、この20年間、毎世代、そのときの半導体技術の限界と、そのときのコンピュータの設計技術の限界に挑戦してきたわけです。僕は、初期のCPUを作ったのは日本人だったし、マイクロソフトが世界のOSを牛耳るのであれば、せめてそれを動かすCPUは日本が作ってもいいんじゃないかなと思ったんです。それで、85年にマイクロソフトとアスキーが提携を解消したときに、日本に帰ってきて、CPUの会社を興したんです。ところが、VMテクノロジーが作ったチップの値段競争が厳しくて、いろいろ手を尽くしてみたけど、結局うまくいかなくて、富士通にデザインの権利を売って、撤退したわけです。それ以外にも、アメリカにアスキーが出資していたCPUの会社がありました。それが「ネクスジェン・マイクロシステムズ」です。この会社でインテルに対抗して高速のCPUを作るということを僕は経験してるわけです。その経験を通して思うのですが、ネクスジェンのCPUはAMDのK5になって、K6、Athronになって、まだちゃんと生きてるわけです。何で日本でそういうことができなかったのかということは、今でも本当に残念に思っているんです。そういうところがハードウェアに対するこだわりとして、潜在意識に残っているんじゃないかと思います。

――なるほど。日本製のCPUと言えば、日立が頑張っていますね。

西　日立のSHが頑張ってるって言うけど、日立はアメリカでアメリカ人を雇ってやってるという話を聞きました。

――ソニーはいかがでしょう？

西　ソニーのプレイステーション2はいいデザインだと思うけど、東芝は自分たちのCPUデザインがもう限界だということに気が付いていて、それでプレイステーション3のCPUはIBMがやってるんでしょう。賢い人たちですね。

――その新しいCPUをネットワークで接続して、というような話を聞きますが。

西　ソニーのプレステ3のチップがどんなものかというのは、インターネットで検索すればいろいろ出てくるけど、どれが本当かはわからない。わからないけど、IBMと一緒にやっているということは、自分でオリジナルなCPUを作るの

を諦めたということじゃないでしょうか。僕が世界で最も速いコンピュータとしてのMSXを作ることになったら、僕もIBMと手を組みたいと思う。インテルと競争できるのはやはりIBMです。今の32ビットのPowerPCを作っているのはモトローラだけれど、PowerPCの未来というのは、モトローラのアーキテクチャというより、むしろIBMの64ビットのアーキテクチャだと思うんです。その、64ビットのインストラクションのサブセットを作って、グラフィックスもキャッシュメモリも1チップにすればいいのにね。IBMのeサーバーの上位機種がマルチプロセッサやグリッドって言ってるけど、PS3もあんな風になるのかなあ…。

1チップ化でMSXが復活!?

── ハードウェアとしてのMSX復活はあるのでしょうか？

西　今の集積度を考えると、ビデオチップとCPUは1つのダイの上に載せるのが正しいと思うんです。ビデオのインストラクションセットとCPUのインストラクションセットが1つになる。そうするといろいろな入力をDSPで管理して、SMTを使ってビデオのインストラクションセットが付いて、MPEG再生とか、タイリングとかできると思うんです。1チップに結局行き着くんじゃないでしょうか。おそらくソニーはすごく進んだCPUとビデオシステムを作ると思うんですけど、今ソニーが作ろうとしているハードウェアは独占でしょ。僕は、未来のデジタル機器、たとえば、未来のテレビ、未来のビデオ、未来のセットトップボックスに、ソニーっていう名前だけしか書いてないという状況は絶対にないと思う。他の会社がそうはさせないでしょう。だからプレステのような機能を持ったもので、誰もが使えるもの。それらを作る人を生み出すために新しいハードウェアを作りたい。ハードウェアと言うけれど、実はソフトウェアのハードウェア。ソフトウェアの乗るシリコン、それを作りたいと思う。それを次のMSXがやるのかといったら、まだそこまでいってないけど……。

── ソフトウェアのハードウェア……、ですか？

西　それがこれからやりたいことです。単にMSXのCPUとビデオとメモリとを1チップにするだけだったらPentiumみたいなものを1つ作るくらいのもので、それだったら古くなったらおしまいじゃないですか。そんなものは作りたくないわけです。何を作るかというと、フィールド・プログラマブル・ゲートアレイ（FPGA）というのがあって、それはロジックを、ソフトウェアをダウンロードしてくることで作れるんです。このプログラマブルなゲートアレイにRISC CPUとビデオプロセッサを付けて1チップのシステムを作る。これなら1万円以下の価格で、ビデオプロセッサとCPUを搭載した、ちょっとしたシステムが簡単にできる。もちろんブラウザもメールも使えます。もう1つ今考えているのは、CPUの動作は、SMT（Simultaneous Multi Thread：同時マルチスレッド）といって、ハードウェアによってスレッドが同時に複数個存在するプロセスが重要だと思うんです。つまり、RISCのインストラクションセット。あと複数のDSPの機能が同時に動くようなマイクロプロセッサ。入力と出力のところでモデムとか、無線の部分でソフトウェアラジオとか、そんなことができるわけです。そうすると、あらゆるネット利用が、全部1チップでできてしまう…。

── なるほど。ソフトウェアのように、目的に応じてロジックを組み替えられるハードウェアということですね。

西　そう。別にOSは限定しないでおこうと思うんですが、Linuxも動くし、その他のOSでも動くようにする。マイクロソフトのPocket PCも……。そうすればみんな自分の好きなようにプログラムをすると思うんです。そのプログラムしたコンピュータのスペック、イメージをネットワークで送り受けをする。そういうふうにしたいと思うんです。

── と、言うことは、MSXのロジックをダウンロードすれば、ハードウェアとしてのMSXも蘇りますね!?

西　そういうこと！

電話にもなる1チップMSX！

西　あともう1つは、これは次世代MSXと関係ないんですけど、インターネットについても考えているんです。ネットワークが高速になっても、つながらない時は全然つながらないわけですね。だから100Mbpsのネットワークとか引いても、ぐずぐずきてるわけ。あれは勘弁してほしい。

── 帯域保証じゃないということですね。

西　だから僕は帯域保証をしたインターネットのネットワークトポロジっていうのも同時に考えたいと思っているんです。そうすると何が可能になるかというと、インターネットで電話とテレビの放送、ビデオストリーム、オーディオストリームが最低のクオリティであっても絶対につながるということなんです。これは、これからの非常に大きなテーマだと思っています。今、どこかのサーバに皆が殺到したらパンクするわけです。だからサーバにデータを取りに行っても、サーバがパンクしないトポロジ。それから、このネットワークに行けば絶対に帯域保証されているというネットワークトポロジ。あんまりスピードが速くなくてもいいから、128kbpsとか512kbpsとか1Mbpsのスピードだったら絶対にデリバリーされるというクオリティ・オブ・サービスが保証される。それらと、このチップが組み合わさった時に、それは非常に面白いことができると思います。

── 具体的には？

西　例えば、チップ1つは1000円とか2000円でしょ、これだけで電話ができるわけ。ワイヤレスの発信と受信ができるわけ。これって電話じゃないですか。今、携帯電話って、ドコモが日本で一番儲かっている会社みたいなもので、料金が高いわけです。携帯電話が月500円になったらどうします？

── それは安い！　電話かけまくります（笑）。

西　いっぺんPHSネットワークができあがってしまった後で、会社を潰して二束三文で買い取って

32kbpsのデータ通信つなぎ放題月々3000円だって。あのネットワークがクオリティ・オブ・サービスを保証したらすごいでしょ？月々3000円で使い放題電話ができるわけでしょ。でもなぜ使い放題電話をしないのかっていったら、そんなことしたらあの会社が潰れちゃうから。32kbpsでも音声のコーデックが動くわけです。ということは、あのPHSのバックボーンをちょこちょこちょこっといじって、32kbpsのスピードでPeer to Peerのコネクションができたら電話になる。月々3000円で日本国中電話かけ放題だったら便利ですよね

――そっちに加入して、思わず長電話しちゃいます。

西　だろ？　そうなったときに、それをサポートするような機能も入れたいなと思っています。この11月から050っていう電話番号が登場したけど、CATV会社が050のサービスをどう言っているかといったら、「ケーブルの加入者の中だけではタダ」っていうことです。ケーブル会社同士が連携する動きがあって、ケーブルテレビ同士だったら、どこまでもタダでいきましょう、と言ってる。お金を払うのはNTTを経由する時だけって。だから今までのインターネットに、＋αのインターネット、この小さなチップによって本当にいろいろなことができるようになると思っています。

――それは、10年前には思いも付きませんでしたね。実は、MSX MAGAZINEは1992年夏に出た号が最後でして……。

西　10年間お休みしてたのね。

――そうなんです（笑）。その最後の号で、やはり西さんは「MSXは続ける」とおっしゃってました。それから「MSXを1チップ化する」とも。つまり、10年間信念を貫き通されてきた。

西　そんなことないんです。迷ったり、リストラとかいろいろあって（笑）。死んだ振りしてたし…。でも、まあ、なんとかここまで来ました。僕1人だけでやってたわけじゃないからね。みんなのお陰です。ありがたいことです。

――高く跳び上がる前に一度しゃがんだ。

西　MSXは16ビット時代をお休みしてたということですね。8ビットから32ビットに行くジャンプのシナリオが32ビットの上で動く8ビットのエミュレータだったってことです。

――そして、1チップ化へとつながる。ところで、チップに実装されるものとしては、CPUとDSPと、それ以外には何が載ってきますか。

西　無線LANですね。

――そう言えば、インテルの新しいノートPC用のチップセットには、無線LANが搭載されるようです。

西　無線LANやるなら、DSPを載せてパケット化とソフトウェアラジオと両方やらないとね。普通に考えたらDSPが2つ。デジタルの部分以外にアナログの部分とRFのチップが必要なんだけど、結構それはコスト的に大変です。

――先ほどのVoIP（Voice over Internet Protocol：インターネット電話の技術）はアナログで？

西　VoIPは全然難しくない。無線のパケット化をどうするかなんです。（試作品と思われる基盤を手に）これがそうなんだけど、これ、ザイリンクスのFPGA、ここにDSPのアーキテクチャを持ったCPUをプログラムして移して、アナログのRFのチップが追加で2つぐらい必要です。この2つのチップを付けるだけでできちゃう。全然問題なくできます。今、無線LANの会社はみんなAT&Tのチップ買ってきてカード作ってるからわかんないんです。ソフトウェア無線のことを基本から知っていたら全然違うということ。

――おぉぉぉっと！　すでに1チップMSXマシンの試作基盤が存在していたとは！！

PROFILE

西　和彦　1956年生まれ。早稲田大学在学中、1977年に（株）アスキー出版（現：（株）アスキー）を設立。日本初のパソコン誌「月刊アスキー」を創刊。1979年米国マイクロソフト社極東担当副社長、翌年ボードメンバー兼新技術担当副社長。近年は国内外の大学など、教育分野で活躍。専門分野は「メディアシステム論」「ベンチャー経営学」「国際政治学」など多岐に渡る。マサチューセッツ工科大学客員教授。博士（情報学）。

1チップMSXはアキバで買える!?

――この1チップMSX（MSX以外のハードにもなってしまうけど）の具体的な製品イメージはどういうものになるのでしょう？

西　まだ最終的に決まってないから、キーボードの中に入れてパソコンにしたり、携帯電話の中に入れていろんな使い方があるでしょう。1チップだから。ユーザーがソフトウェアでいろいろな使い方ができるような、そんなチップです。

――アキバなんかに売ってて、買ってきて自分で組み立てて、プログラムもいじってみようかなんていうのは、ありですか？

西　そうそう。レゴみたいな感じ。

――素晴らしい！　電子ブロックも復活していますしね。

西　電子ブロック復活してるけど、何作るの、あれで？　お湯が沸いたらピピピっていうの、もうお風呂に付いてるよ。でも、みんなが電子ブロックを求めているというのは、自分でハードウェア作りたいってことでしょ？　プログラムだけじゃ、いやだ。バーチャルワールドのロジックだけじゃ、いやだってことだと思うんです。

――そこで1チップMSXの登場ですね！　ずばり、登場時期は？

西　それはまだ決まっていません。プログラムゲートロジックの設計は動いてたりします。DSPとビデオのあたりを一緒にするようなこともやっていますから、あとはまとまったお金が……。PLAYerが売れてアプリケーションが売れて、マスクデザインを発注できる1億円ぐらいお金ができてから……。スポンサーを見つけてこなきゃね。前みたいにお金が沢山使える立場じゃないから。今は台風がきたら雨が漏るような所の事務所にいますから（笑）。だから稼がなきゃならんのです。

――みんながMSXPLAYerで遊んだり、勉強したりすればするほど、1チップMSXの登場時期が早まるというわけですね。本日はありがとうございました。

すべてのきっかけはMSXだった

古谷徹さんと言えば、幅広い年齢層にファンを持つ人気声優。古谷さんの声さえ聞けば、星飛雄馬やアムロ・レイなどの顔が思い浮かぶんじゃないかな。名車のビジュアルが浮かぶ人もいるかもね。そして対するは、MSX MAGAZINEの表看板、「たぐっちゃん」こと田口旬一初代MSX MAGAZINE編集長だ。実はこのお二人、古くからのお知り合い。そこで、MSX MAGAZINE復活を記念して、素顔のお二人にMSXの思い出を存分に語ってもらった。

何から何まで型破りな創刊

――MSXとの最初の出会いというのは、どういったいきさつで？

田口 俺なんて、出会いも何もないよ。もともとアスキーでMSXの雑誌を作るという話があって、塚本さん（当時アスキー副社長）から、「編集長が決まったら、君がバックアップしてくれ」という話があったの。その時、書籍の次長かなにかをやっていたかなぁ。で、ある日、その塚本さんに「編集長が決まったから、ちょっと来てくれる？」と呼ばれたの。そこで「次の編集長誰ですか？」ってきいたら、「君！」って言われて、ハイおしまい（笑）。その一言。で、俺は「いやです」って言ったの。そうしたら、「お前、一度断ってるからダメ」って言われた。いや、実は一度「LOGIN」の副編集長を断ってたのね。

古谷 何でいやだったの？

田口 その時は雑誌の編集をやったことがなくて、すごい不安だったの。書籍編集部も立ち上がってきたばかりのときだったので、俺が抜けちゃうと、書籍の方の責任者がいなくなっちゃうでしょ。ところが、他の編集長もみんなこの話を断ってて、塚本さんも困り果てて「もう、こいつしかいない」ということになって、俺に編集長をやれ、と。たったそれだけのこと。出会いもへったくれもなかったよ。業務命令だから。やりなさい、と。

――無理やり出会わされたわけですね。

田口 で、LOGIN編集部にいた編集者と俺の2人で始めた。

――たった2人ですか？

田口 2人。

――すぐにスタッフは増えたんですよね？

田口 いや、増やすのは新卒の社員を使えって言われていたから、翌年の4月までは2人のまま。結局、俺が編集長を辞めるときまで、6人を超えることはなかった。それ以上増やしてくれなかったからね。いつ潰れるかわからない編集部に人なんかやれるかって（笑）。

古谷 うっそ～ぅ!?

田口 いやいや、嘘じゃなくて。だってMSX立ち上げの時点では、本当に売れるのかどうか、まだわかってなかったし。西社長だけは張り切っていたけど。それに、俺もそれまで雑誌なんかやったことなかったわけだし。その代わり、型破りだったな。だって最初縦組みだったしね。当時、縦組みのパソコン誌出すなんて、無茶だって。

――今でこそ「週刊アスキー」とかがやってますけどね。

田口 やってるでしょ。中綴じも初めてだった。そのときのことはよく覚えてるけど、ただ単に普通の作り方してもインパクトがないから、縦組みで作ってみようっていう話になったの。しかも、ほとんどのコンピュータ誌が平綴じだった。で、もっと一般的で柔らかいイメージってことになると、中綴じにした方がいいんじゃないか、と。

古谷 そういえば、当初はそういうデザインでしたよね。

田口 大きさもAB判っていうサイズだった。昔、アメリカのカタログ誌とかがAB判でね。書店店頭に並んだときに、他の雑誌の中に埋もれちゃいけないっていうんで、AB判にした。だから、パソコン誌の慣例を、次から次へと打ち破って作っていたのね。

――画期的だったんですね。

田口 いや、画期的というより、単に型破りだった。

古谷 創刊号は83年6月？　そんなに前だったんだ。あ、そっか、MSXが発売される前からMSX MAGAZINEはあったんですね!?

田口 ありました。2号までは実機が発売される前でした。製品が出る前に雑誌作んのぉ～？って思った……。

古谷 そうだよねぇ。実機がなかったんじゃあねぇ。

—— でも、さすがに仕様は決まっていたんですよね？

田口 いや、創刊号の時点では、まだ仕様が決まってなかった。で、西さんが「俺が記事を書く」って言って書いてくれたの。でも、西さんが書いてくれた原稿は、MSXの暴露記事みたいになってて、これはまずいよって。こんなこと書いちゃダメ、みたいな（笑）。もう、ボツ、ボツ、ボツ。そうしたら、西さんが怒っちゃってさあ、でも、怒ったってこんなヤバイ話は載せられる訳ないでしょうって言って、ボツ（笑）。

—— 社長の原稿をボツですか。まさに型破り！

古谷 でも、西さんのアスキーだからこそできたんだよね。じゃなきゃ、できないですもんね、実機がないのに雑誌を出すなんてことは。

楽器として買ったのにBASICにはまった

—— 古谷さんのMSXとの出会いはどういういきさつで？

古谷 僕ね、84年の1月なんですよ。ヤマハのCX5を買ったのが。

—— それ以前にパソコンは触られてました？

古谷 いや全く。スラップスティックっていう声優で作ったバンドをやってて、年に2回ぐらい大きいコンサートツアーをやったり、レコーディングをやったりしていたんですが、ちょうど84年の1月に有楽町の読売ホールでお正月のコンサートがあったんです。僕はドラム担当なんですけど、ステージ上に3メートルぐらいのやぐらが組まれてて、その上でドラムたたいてたんですよ。我々のコンサートっていうのは、メンバーがみんな役者なんで、2部構成にして、曲を演奏した後、芝居もやってたんですね。だから、第1部が終わって、舞台が暗転している間に1度楽屋に戻って衣装替えをしなくちゃいけない。その時にドラムの台から下りようとしたら、そこにあるはずの梯子がなかった。なのに、足を踏み出してしまったんですよ。で、落っこちゃったんですよ、3メートル下へズドンて感じで。それで、僕は肋骨を折ってしまったんです。コンサートの本番中ですよ。

田口 うわぁ。それは大変だ。まだ次の第2部があるんですよね？

古谷 もちろん。それで急遽、どん帳を下げて、休憩にしちゃった。僕は苦しくてしようがなくて、小さい声で「救急車」って言ったんですよ。そうしたら救急車が来て、お客さんも見ている中を担架で運ばれていっちゃったわけです。コンサートは中止でした。さらに1カ月間は静養しなくちゃいけない。肋骨だから声が出せず仕事はできない。いきなり、1カ月暇になっちゃったなぁ……、と。でも歩けるんですよ。コルセットして病院に通ってましたから。そのときに、「Player」という音楽雑誌を読んでいたら、そこにヤマハのCX5の広告が出てたんですよ。

—— 音楽雑誌ですか。

古谷 画面に音符が映ってる。これ、テレビに音符が映ってるぜ、みたいな感じで新鮮な印象だったわけです。これはどうやらコンピュータらしいってわかったけど、パソコンなんて俺きっと使えないだろうな、無駄遣いになっちゃ、もったいないなぁ、とも思ったんです。でも、よく読むと、これはシンセサイザーになるって書いてあった。パソコンはわからないけど、シンセサイザーならオモチャになるからいいじゃない、と思った。とにかく画面に音符が出るというところにすごく惹かれて。ちょうど暇になったことだし、値段も59,800円と安いし、じゃあ、これ買ってみようということになって、いざ買ってみたら、夜ごとBASICのプログラミングにはまる毎日。

田口 それは凄すぎる。

古谷 すっかりはまっちゃった。

田口 楽器として買ったはずなのに。

古谷 そうなんですよ。もうBASICが面白くて面白くて。

田口 当時、シンセで始めて、BASICにはまる人って、あまりいなかったですよね。

古谷 そうですね。ただ、はまる素養というのはあったんです。僕は大学の時に、COBOLとかFORTRANでのプログラミングを一応習っていまして。当時は、パソコンなんかありませんから、マークシートのカード束ねてプログラミングしてたけど。だから、BASICでいきなりPRINTって打つと文字が出るということに感動しちゃって、いじり始めたら、すっかりはまっちゃいましたよ。それで最初インベーダーゲーム作ったんです。自分で。

—— いきなり凄いのに挑戦されましたね。

古谷 いやぁ、試行錯誤しながらですけど、それが面白くって。それでいろんなプログラムを次から次へ作り始めてましたね。（自作プログラムのファイルを見ながら）このプログラムを組んだのは、1984年の5月ですから、いじり始めてまだ4カ月目ぐらいなんですけど、要するに和音のコードを解析して演奏するプログラムでしたね。これヤマハのMSXでしかできなかったんですけど、コードネームを打ち込むと、その和音が出る。逆に鍵盤で和音を弾くとコードネームを教えてくれる。（ファイルから1枚取り出し）これがコードの構成和音表です。

—— きちんとファイルされているあたり、まめですねえ。

古谷 いや、僕もそれまでそうでもなくて、まさかこんなにはまるとは思っていなかったんです。でも、考えてみたら英語も数学も大好きだったなと。それに、これって悪いのは自分なんですよね。エラーが出るのは、必ず自分のプログラムにバグがあるわけですよ。それのどこが悪いのか、探すのがパズルのようで楽しくて楽しくてね。

—— 自分への挑戦ですね。

古谷 そうそう、自分への挑戦。本当に楽しかったですね。

パソコン通信が縁で出会った

—— お二人はそもそもお知り合いだったそうですが、やはりMSX MAGAZINEがきっかけですか？

古谷 そうですね。なんでだろう。どっからなんだろう。何からでしょうね。僕もよく覚えていないんですけど。

田口 古谷さんがうちの雑誌の読者だということを誰から聞いたのかなあ……。MSX MAGAZINE読者に有名な人がいるって聞いて。

古谷 たぶんネットじゃないかな。僕、アスキーネットの実験開始から参加してましたから。アスキーがパソコン通信を始めるっていうんで、そのためのムックが発売されて買ったんですよ。PC-9801か何かの通信プログラムが掲載されていて。僕、それをMSX BASICに移植したんです。それでMSXで通信を始めたんですよ。それからじゃないかな。それでMSXのフォーラムとかそういうところに書き込みしたりしてたんで、それがき

対談にも登場した古谷徹さんの声が聞ける、お手製Windowsプログラムは、古谷さんのホームページからダウンロードできるぞ。

http://www03.luvnet.com/~pegasus/

PROFILE

田口旬一：元MSX MAGAZINE名物編集長。本誌付録シールにもある「たぐっちゃん」の絵のモデル。現在は、株式会社ぴえろ勤務。

っかけじゃないかな。そこで編集部の人と知り合ったんじゃないかと。当時、夜中にネット上にいる人って、コンピュータ業界の人か、出版業界の人しかいなかったですよね。

田口　そうそう、あのときは確かに「古谷さんこんばんは」とか言いながらやってた。あのころ300ボーしかなかったの。よく回線が切れてね。

――しかもMSXで通信されているとカタカナですよね。

田口　そうそう、カタカナ。

古谷　最初はアルファベットだけだったからローマ字で打ってたもん。読みにくいったらありゃしない。で、少しだけましになったのがカタカナですもんね。結局、カタカナも読みにくかったですよね。漢字が使えるようになったの、ずっと後になってからだもんなぁ。

田口　1200のカートリッジが出てからですね。

声優を辞めてSEになろうと思ったことも

古谷　僕、自分のコンサートで、自分で作った曲をMSXに打ち込んで歌わせるようにして、それをライブハウスに持ち込んで一緒にデュエットしたことがありますよ。ヤマハのCX5のFM音源は、優れてたんですよ。カタカナで言葉を打ち込むじゃないですか、それに音程を付けることができたんですよ。そうするとその音程で、ちゃんと歌ってくれるんです。同時にピアノの5声ぐらいの和音も弾いてくれちゃう。当時は珍しかったから、お客さんみんなびっくりしてくれましたね。

――当初の目的通り、音楽方面にはかなり活躍したんですね。

古谷　ええ、活躍しましたね。

――それ以外には、実用にも使われたんでしょうか？

古谷　何と言っても税金の経費計算プログラムは凄く役に立ちましたね。MSXの元手は充分に取れました（笑）。税金プログラムを最初に作ったの1985年7月だ。

田口　85年はちょうどMSX2の登場した年ですね。

古谷　しかも「Tax for Disk」としているのは、その前にテープバージョンも作っているんですよ。ディスク用に作り替えたんです。もう18年ぐらい前ですか。その時に税金をパソコンで計算していたという。

田口　すごいですよね。しかも自作プログラムですから。

――今だったらExcelとかでやっちゃうんでしょうけど。

古谷　そうですね。僕も今はExcel使ってますけど。

田口　当時は、無いものは自分で作るのが当たり前でしたよね。

古谷　そうそう。必要なものは自分で作ってた。プリントアウトして税理士さんの所へ持っていったりしてましたね。お遊びで人工知能作ったり、自動作詞家なんかも作ってましたね。これはMSX2用ですね。もう漢字が使えてた。自分で作詞したフレーズをいくつか入れておいて、それをランダムに組み替えさせて、勝手に作詞をさせるという。あ、これは番号照合プログラムですね。これも元はMSXですけど。お年玉の年賀ハガキの当選番号を照合するプログラムです。今でも年賀状が1000通近く来るんですよ。もしかしたら1等当たってるかもしれないじゃないですか。それを1つ1つ見るのは大変だったんですよね。なので、これはプログラム作っちゃおうと思って、下1桁目から順に打ち込んでいくと、はずれの時点でピッて鳴るようにしたの。ピッて鳴らない限りずっと打ち込んでいく。最後まで鳴らずに打ち込めれば、その番号が当たっているっていうことなんです。それは未だに使ってますよ。Windows版になって。今は僕の声で「大当たりだじょぉぉっ！」ってしゃべるようになったんですけど（笑）。

田口　スゲー。それはすごい（笑）。

――ぜひ、聞いてみたいです。

古谷　僕のホームページで公開されてますよ。ダウンロードして遊んでください。それをちょっと改造して、宝くじの照合プログラムも公開していますけど。今公開してるやつはタイマーとか、簡単なやつですね。それもやっぱり「時間です」ってしゃべったりするわけですよ。カーグラTV調の声で（笑）。基本的に僕がしゃべるプログラムを公開しています。

――今でも自分でプログラムを組まれるのが楽しいですか？

古谷　そりゃ、楽しいですよ。ただ時間がないとできないじゃないですか。（自作プログラムのファイルを見ながら）これなんかもそうですね。作ったのが1月1日から1月4日になってるから、たぶんお正月休みでしょう。休みなんですよ、集中してできるときって。MSXでコンピュータの世界に入ってから、いろんな方面に興味持ち始めました。一時期、情報処理技術者の資格取ろうかなあとか、SEになろうかなあとか、声優をやりながらも思っていました（笑）。

――えぇっ！　古谷さんが職業プログラマに転身ですか？

古谷　こんな面白い仕事があるって、だんだんわかってきて。やりたいなあってずっと思ってたんですけど、幸い仕事にも恵まれてしまったので、結局、声優辞められなかったけど。

田口　あぁ、よかった（笑）。

古谷　でも、そのおかげで、未だにこの業界で、コンピュータでわからないことがあれば古谷に聞けばわかるみたいな状況になってて、今はホームページの先生になってますけど（笑）。みんなのところに行っては教えてあげたりして。

田口　すべて、きっかけはMSXだったんですね。

古谷　もちろんそうです。パソコン通信があったからインターネットにも入りやすかったしね。

お母さんが始められるコンピュータ

田口　思い出に残るいい話があっ

てね。子を持つお母さんから編集部に電話がかかってきて、画面が出ないんですけどって。うちはメーカーじゃないのに、よろず相談所かって(笑)。で、テレビの前にMSXを置いて、両方とも電源入れたけど映らないと。で、MSXとテレビは何でつないでいますかって言ったら、「えっ?」って。「どうやってつなぐの?」って。「アンテナかビデオ端子で」って言ったら「早く言ってよ」だって。

古谷 ちゃんと取説読まないでやっちゃうんですよね。

田口 でもね、お母さんが始められるコンピュータっていう意味ではすごくよかったと思うわけ。

古谷 ああ、そうですよね。

田口 しかも、テレビの前にコンピュータ置くだけで使えるっていう発想がすごいじゃないですか。

古谷 そうだと思っちゃったんだ。

田口 思っちゃったらしい。電気屋さんに行って買うのはいいけど、小さいから持って帰れちゃうでしょう。持って帰ってきてスイッチ入れたら点かなかったと。

古谷 MSXって、安く買えるという意味で、やっぱり入りやすかったっていうのはありますよね。テレビがあればいいんだからね。

――気軽に買えちゃうから気軽に使えると思ってしまうわけですね。

田口 壊れてもいいと思っている部分があったりして。いや、悪い意味じゃなくて。昔はパソコンって高価な物っていう感じがして、壊れたらどうしようっていう恐怖心があったけど、MSXなら多少のことをしても大丈夫、みたいな。

古谷 ニッキュッパとかで売ってたんですもんねぇ。

田口 そういう部分では、ユーザーの意識改革に貢献できたんじゃないかな。それに古谷さんの購入動機と同じで、プログラミングがダメでもシンセサイザとして使えるとか、ゲーム機としても遊べるという形で。これがだめでも違うものにいける気楽さというものがあった。僕らもそういう意味からするとMSXはやりやすかった。コンピュータっていうとプログラムを組むとかになりがちだけど、そういうことじゃなくて、こんなこともできるから面白いでしょう、みたいなことも紹介できたりして。

――CGとかは、解像度がネックでイマイチだったかもしれません。

田口 いや、そんなことはないの。実は大野一興さんみたいなイラストレーターの方たちは、MSXの低い表現力でも、これだけのことができるよっていうのを証明していた。特に、一興さんはMSXがすごく面白いって言っていた。

古谷 大野さんの作品は、毎号表紙を飾っていましたね。

田口 俺が、毎号、ひたすら早く描け、早く描けって首絞めてたんですよ(笑)。ある日、全部描き終わって、やったーって思ったら、電源蹴っ飛ばしてポーンて……。

古谷 ひょえー、できすぎた話ですね。本当にあったんですか?

田口 本当にあったの、あったの。目の前で消えたんだから。それも夜中の3時。次に上がってきたものは、前とは全く違うものだった。大野さんいわく、もう前の作品のことは忘れたって(笑)。

MSXはガンダムと一緒だ!

――MSXPLAYerを実際に見られてどうですか。

古谷 これWindows上で動いているんですよね。フル画面にしなくても、Windows上で何かほかのことしながら、遊べるんだ。いいなぁ。昔、僕がBASICにはまったように、今の子供達がプログラミングに興味を持ってくれると嬉しいですね。

――あとは親子のコミュニケーションにもいいかもしれません。

古谷 あ、そうか。昔、お父さんがやってたからってことね。なるほど。それ、ガンダムと一緒だ。

PROFILE
古谷 徹：声優。代表作に、「巨人の星」星飛雄馬役、「機動戦士ガンダム」アムロ・レイ役など。幅広い層にファンを持つ。

――と、言いますと?

古谷 僕、よくガンダムのイベントに呼ばれるんです。この夏にも「ガンダムワールド」っていうのがあったんですが、そこに来てる人たちっていうのは、いわゆる一時のマニアックな若者たちだけじゃなくって、小学生を連れたお父さん、お母さんが来てるんですよ。家族で。それでお父さんが一生懸命モビルスーツの説明を子供にしてるんですね。ああ、こういう時代になったんだなあって。お父さんはもう意気揚々としてるんですね。子供は子供でゲームからガンダムの世界に入ってるじゃないですか。お父さんはちゃんとしたストーリーも分かっているから、子供に教えてあげられる。ああ、いいなあと思ったんですけど。まさにMSXがそうですね。そうなるかもしれないですね。

田口 最初にガンダムを見てた世代って、最初にMSXを使ってた世代と、近いかもしれませんね。

古谷 ちょうどガンダムの映画の頃かなあ、MSXが登場したのは。83年ってことはそうですね。79年がテレビ放送開始ですからね。そういう時代ですね。そうそう、よく思ったんだ。ガンダムの最初のシーンで、アムロが顕微鏡見ながら「このコンピュータ組んだら食べるよ」っていうセリフがあるんですよ。で、その時に、「えーっ、コンピュータなんて自分で作れるわけないじゃん」って思ってたわけですよ。でも今や、僕も自作のコンピュータ組み立ててるんですけど。パーツ買ってくれば組めるじゃないですか。まあ、顕微鏡は使ってませんけど。

――次はぜひ、ガンダムイベントだけでなくMSXのイベントのほうにもお運びいただいて……。

古谷 え? いまだにMSXのイベントなんてやっているんですか?

――ええ、「MSX電遊ランド」といいまして、昨年そこでMSXPLAYer制作が発表されました。

田口 俺、それに呼ばれてた(笑)。

古谷 知らなかった……。それは楽しみですね!

――次回の「MSX電遊ランド」では、古谷さん、田口さんの生対談が実現するかもしれませんね。

MSXPLAYerで懐かしのゲームを感じ取れ!

プロジェクトEGGは、MSXをはじめとする80年代PC用のレトロゲームを現在主流の環境であるWindows上に再生し提供するプロジェクトだ。MSX用ゲームの実行環境にMSXPLAYerを使っている。しかし、そこに至るまでの道のりは決して平坦なものではなかったはず。そこで、プロジェクトEGGの誕生から、現在、将来までを、プロジェクトEGGを運営するボーステック株式会社に直撃インタビューしてみた。

プロジェクトEGG誕生秘話

――プロジェクトEGG設立のそもそもの経緯からお聞かせください。

鈴木 もともと私は個人的にレトロPCゲームが好きでして、実は入社した動機の1つでもあったりするんですね。で、当時(2001年春頃)弊社は携帯電話にレトロゲームを次々と移植していってたんです。それで他社さんのも含めてうちにはいろいろなレトロゲームが集まって来つつあったんですね。そんな中、ある出会いがあったんですよ。

八巻 うちに「レリクス」というゲームがありまして、これは1984年頃のゲームなんですが、そのPC-9801版を無償公開していたんです。それで、あるとき高木さんという方から、自分のところのPC-9801エミュレータでレリクスを使わせてもらえないかという話がありまして、「いいですよ」とお返事した。

鈴木 この方もレトロPCゲームが大好きでそれでもう意気投合しちゃったんですね。いえビジネスの話ではなくて完全に趣味の領域で。それで話が盛り上がるなかで「コンテンツ集めるから、開発に協力してくれます?」って話になって。ちょうど少し前からうちに「Soft-City」というダウンロードサイトがオープンしてたんです。そこの決済システムを使えばいいってことになって。

――すでに材料は揃っていたわけですね。

鈴木 そうですね。彼も「それなら一肌脱ぎますよ!」と。

――ビジネス抜きのおつきあいからビジネスが生まれた。

鈴木 ええ。でも具体的な話になったのはその後です。2001年の初夏の頃でしょうか。話が具体的になって盛り上がったその日に、私が社に戻ってきて、その概要を八巻に伝えたんです。最初、八巻は「それってどうなの?」と半信半疑で聞いていたんですが、それから私が帰宅するまでの間に、やる気になっちゃいまして。その2日後ぐらいにはもうスタッフが集まって、このプロジェクトをすぐ動かそうということになりました。それで具体的に動き始めたんです。でも引くに引けない状態になってしまったのは、実は「MSX電遊ランド」でプロジェクトEGGのことを発表しちゃってからですね。最初は出席をしぶっていたのですが、電遊ランド実行委員だった横居さんに強引にセッティングされてしまいまして(笑)。

MSXPLAYerが突破口に

――電遊ランドでは、西さんがMSXPLAYerを出すと宣言されましたが……。

鈴木 そうですね。私はその時に初めて西さんとお会いしました。これがきっかけで後に西さんから大きなご協力を得られることになるんです。

――具体的にどんな協力を得たんでしょう?

鈴木 まだMSXPLAYerについてはいつ頃からEGGでサービスを提供していただくか決まってなかったんですが、MSXPLAYerの安定した再現性を見るとやはり公式なBIOSを使用した再生環境というのは出来るだけオリジナルに近い再現を目指す上で不可欠なんですね。で、何とかしてBIOS関係の使用許諾をもらえないかということで、弊社の方から独自にあたってはいたんですけど、なかなかマイクロソフトさんからOKが出なかったんです。

――すべての道はマイクロソフトさんに行き着きますよね(笑)

鈴木 それでダメもとで、ある日西さんのところに強引にお邪魔して、ご相談したところ……。

――よっしゃ、よっしゃ、任せとけと(笑)。

鈴木 ええ、まさにそんな感じで(笑)……。いえ、もうちょっと普通に(爆笑)。

八巻 タイミング的にもちょうど、何日後かにビル・ゲイツさんに会うということでした。

鈴木 それで本当にマイクロソフトさんとお話していただいて、とうとう、EGGで正規BIOSを使ったシステムを組み込んで販売することができるようになったわけです。まさにMSXPLAYerが突破口になったわけです。その後NECさんや富士通さんからもEGGならいいよと快諾を得ることができました。

当時の自分を思い出して楽しむ

――プロジェクトEGG公開後のユーザーの反響はいかがですか?

鈴木 30代、40代の方がすごく懐かしがってくださいます。「あの当時のゲームで遊べる環境を作ってくださってありがとうございました」という激励の言葉もありました。また、やっと手軽に遊べるようになったというご意見もあります。逆に「何であのゲームは出ないのか」と叱られたり「あのメーカーを参入させてください」とか強い想いのこもったリクエストが多数届いています。

八巻 やはり正規に遊べる環境を提供したということで反響が大きかった。

鈴木 うちの会員は詳しい方が多いんです。私よりもずっと詳しい。それとプロジェクトEGGの意義をとても理解して応援してくれている。もうお客さんじゃなくて、まるでボランティア精神みたいな。ほんとうに暖かく積極的に援護してくれる。ありがたいです。まさに同じ趣味を持つ者同士、どこかで気持ちが繋がっている。

――やはり、30~40代が多いのでしょうか。もう少し若い世代は?

鈴木 たぶんその年代の方ってゲームだけが面白いというんじゃなくて、当時の自分を思い出していて、それも楽しんでいるんじゃないかな、っていうのがあります。

八巻 まだ若い世代のユーザーへのアピールが足りてないというのはあると思います。また、知っていても入会できる環境ではないというのもあります。決済方法がクレジットカードなので、それがネックになっています。現在、BIGLOBEさんの方でも、決済サービスを行っているんですが、それだけではまだ……。

――画期的な決済手段が出てくれば、年齢の広がりも期待できますね。

八巻 そうです。今後、携帯電話まで視野に入れて決済サービスを拡大していくとかそういう話もありますし、とにかく選択肢を増やしていきます。

鈴木 ゲームショウでの反応を見ると子供達も予想に反して驚くほど自然にレトロゲームを楽しんでますよ。単純に画面だけで見れば、今はもっと凄いグラフィックスのものがあるわけですから。もっと年齢層が広がる可能性は充分にあると思っています。

――親子でMSXとか(笑)。

鈴木 いいですね。レトロゲームで親子の絆を深める(笑)。

ゲームアーツも参入!?

――現在、何タイトルくらいダウンロードできますか?

鈴木 現在販売しているタイトル数は約140種類ですね。

――そのうちMSXのタイトルは何タイトルぐらいありますか。

鈴木 今は、24、5だと思います。この本が出る頃にはゲームアーツさんも参入していると思います。

――それは楽しみですね。現在毎月平均10タイトルがアップロードされていますが、このペースは今後も続きますか。

鈴木 可能であれば10タイトルにこだわらず、それ以上に配信したいと思っています。最近仕事の流れができて安定してきましたので、いずれ可能になるのではないかと思います。

MSXゲームをPDAや携帯電話へ

――プロジェクトEGGの今後の展開はどうなりますか?

鈴木 実はX68000についてもシャープさんからBIOS等を提供してもらうことになっています。ですので、PC系はこれでほぼ出揃います。

八巻 ほかには、コンシューマゲーム機もやっていきたいなと。権利の問題がいろいろとあるので、仕様等詳細はまだわかりませんが。

――現状ではWindows上での提供となっていますが、プラットフォームを増やしていく予定は?

八巻 Linuxが普及するのであれば、載せていくかもしれません。それと、PDAや携帯電話への移植はMSXで進めていると聞いています。こちらとしてもそういった動きには積極的に(ゲームコンテンツの用意や契約の整理、決済システム等で)対応していきます。まだ可能かどうかわからないんですが、GAMECUBE、PS2、Xboxのようなコンシューマ機でもいくつかの条件を満たせば、ネットワーク配信することもできるのではないかと思っています。

――最後に何かあればどうぞ。

八巻 現在プロジェクトEGGへの協賛ソフトハウスが25社ぐらいになりましたが、まだまだ参入していただきたいので、もし興味のあるソフトハウスの方がいらしたら、ご連絡をください。

鈴木 もっともっとゲームを増やしていきます。みなさんぜひレトロPCゲームを楽しんでください。リクエストも受け付けますのでお気軽にメールしてください。スタッフ一同、趣味と仕事を兼ねて(笑)頑張っていきます。よろしく!

――本日はどうもありがとうございました。

Soft-CityにあるプロジェクトEGGのWebページ。ココがMSXをはじめとするレトロPCゲームがオンラインで購入できるサイトだ。本誌付録CD-ROMに収録されている体験版「イース」「イースII」「ワンダラーズフロムイース」の製品版もプロジェクトEGGで購入できるぞ。なお、同サイトで購入可能なMSXのゲームの一覧はP.222で紹介しているので要チェック!

http://www.soft-city.com/egg/

いつでもどこでもが僕のMSX 3

アスキーで10年もの間MSXの開発に携わってきた山下氏こそこの世の中でもっともMSXのことを知りつくしている人物である。編集部では、"公式"と銘打つMSXPLAYerに山下氏が関係しているという情報を得て、さっそくお話を伺うことにした。当初は、MSXPLAYer＝MSXエミュレータという単純な構図を想定していたのだが、伺うにつれ、その意味の重要性に驚かされることに。MSXPLAYerは、"アプリケーションプラットフォーム"という理想を実現するための第一歩となるかもしれないのだ。

アスキーブランドのMSX

――MSX(実機)の終盤は、山下さんとしてはどんな心境でしたか。

山下　MSX3といったら語弊がありますが、もう少し先のことまでやってたんですよね。結局メーカーさんの開発スピードとカミ合わなくなっちゃったんですけど。状況からすると「寂しいけどもしょうがないね。それが世の中の流れだろうね」という感じでした。しかし、ニーズはゼロではないはずだから、メーカーさんが全部撤退した後は、いい商売ができる可能性があるとも考えました。物を作ることに関してはメーカーさんに前面に出てやってもらうのが原則だと思いますし、アスキーとしてはある意味黒子に徹していないといけませんが、実はアスキーブランドのMSXも検討したことがあります。夢ですよね。そういうのもできるとおもしろいなと思ってました。ただ、現実的に物を在庫することにはリスクが伴いますので、そこは検討で終わりましたけど。

――それはすごい話ですね。うーん残念。MSXには都合どれくらい関わっておられましたか。

山下　10年ぐらいです。立ち上げたのは83年ですが、その前から作ってましたから。

――最初からアスキーにいらしたのですか。

山下　正社員として入社したのは82年で、その瞬間からシアトルのマイクロソフトで日本語版MicrosoftBasicの仕事をしてました。ちょうど16ビットパソコンが出てきた頃です。シフトJISも作りました。自然に、いろいろなパソコンに関わるようになって、それで「もう次から次からパソコンを作るのは嫌だ」と西(西和彦)さんに言ったら、これが最後だからと言われて10年も続いてしまったのがMSXだったという(笑)。

――マイクロソフトとしてはMSXはどんな扱いだったんですか。

山下　マイクロソフトでは、すでにIBMとの話が始まっていて、16ビットの方がメインストリームになるとわかっていました。8ビットにいつまでも関わっていたのはアスキーから出向いていた僕らだけだったんです。だから「アスキーは好きにやれば」という感じでしたね。マイクロソフトは手は出さない代わりに手助けもしないよ、と。

――それでMSXは"マイクロソフトのもの"という感じがあまりしないんですね。

山下　僕らもそうでした。ただ、マイクロソフトには足かけ3年いたんですけど、マイクロソフトのBasicとか、マイクロソフトの開発のノウハウを学習して、それを総合して作りあげた物がMSXだということで、そういう意味ではちゃんとしてるんです。だから、あの当時からプラグ&プレイですよね。カートリッジをガチャンとさしたら動くという。日本語をサポートするためにやるべき手も最初から打ってありました。それと、他のPCはたいてい裏側に拡張スロットがあるじゃないですか。そこは、簡単には触れられない領域ですよね。しかし、それではおもしろくないので、MSXのスロットは常にユーザーの方に向いていて、すぐに抜き差しできるようにしました。これなどは今振り返ってもいいんじゃないかと思います。

プラットフォームはインテント

――MSXPLAYerの話はどこから始まったんですか。

山下　MSXPLAYerに関しては、後から巻き込まれたんです。僕がそもそも関わっていたのはMSXPLAYerのプラットフォームになっている「インテント」です。すごく個人的な話になるんですが、MSXはおもしろかったんですよ。

10年もやれて。ある年、MSXの新機種が出ないという話になったんですが、またこうしたものをやりたいなと思ってました。そうこうしているうちに、突然西さんに「おいお前、すごいOSがあるから来い」って会議室に呼び出されて、行ってみたらインテントでした。西さんが見つけてきたんですよね。特徴はというと、CPUは何でもかまわない、OSも何でもいいという。それまでにも、ソフトウェアがOSやCPUに依存しない環境というのはどうしたらいいのか、という議論はいつもあったんですよ。それは、MSXをずっとやりながら遂に果たせなかった夢でした。インテントならそれが実現できるのではないかということで96年ぐらいから関わって育てていました。「パソコンはWindowsでいいや」と思いますが、携帯電話やPDA、あるいはテレビやカーナビまで全部Windowsということは考えられないですよね。かつ、そういう世界ってCPUが全部違うじゃないですか。必然的にインテントみたいな仕掛けが求められるはずだと思ったんです。

――西さんもですか。

山下 PCでは当時、WindowsとMacが両方がんばっていたので、西さんは「こういう仕掛けがあれば、双方をブリッジできる」というコンセプトですごく入れ込んでいたようです。私はどちらかというと、いわゆるエンベデッドな組込用の機械の方に使う仕掛けとして、きっと世の中に求められるに違いないと関わり始めました。それで99年頃かな、インテントの完成度が上がってきたので「MSXPLAYerで使ったら」と提案しました。ですから、単にMSXエミュレータが動けばいいというだけなら、いきなりWindowsの上で動いているようなものだったでしょうね。

――PocketPCでも動いている理由はそれなんですね。インテントが非常に興味深いプラットフォームだというのは分かりましたが、OSを選ばないという環境としてはJavaがありますよね。

山下 一番簡単な言い方をすると、JavaはJavaしか使えないじゃないですか。C言語では書けません。でも、インテントはCPUのレイヤー(層)で仮想化しているので、C言語でもアセンブラでもJavaでも書けます。つまりJavaよりもずっと適用範囲が広いわけです。インテントから見ると、Javaは"One of Language"なんです。それと、ゲーム開発者から見るとJavaはあんまり嬉しくないんですよ。C言語に慣れた人はCで、チューニングしたくなったらアセンブラでガリガリ書ける環境が欲しいということもありますし、一番困るのはJavaはメモリのお掃除(いわゆるガベージコレクション)しますよね。あれが起きると秒間何コマの速度が落ちます。しかも、いつ起こるともしれません。そういうことで、つい口を出してしまってインテントでやることになりました。

MSXPLAYerの評価は

――実際に開発の話をしていただきたいのですが。MSXのエミュレータをインテントで動かすというのは、割とすんなりと進んだんですか。

山下 そういう話になると私は野次馬で、横でこうしたらとかああしたらとか言ってたわけですよ。実際に作業されていたのはMSXアソシエーションです。開発者が困ったときにアドバイスをした程度です。エミュレータを作るには完璧にハードウェアのことをわかっていなければならないわけですが、実際は細かいレベルまで知っている方が結構いたりして、あんまりそういう面ではお手伝いしてないような気がします。「インテントでやったら」と言ってしまった責任もありますし、不純な動機でいえばインテント・ビジネスの役に立つだろういう立場もあってお手伝いしていたわけです。いつも返答が遅いとかって怒られていましたけどね(笑)。

――ということは、山下さんが関わったときにMSXPLAYerはある程度動いていたんですか。

山下 そうです。これを"公式"の名に恥じないようにキチンと完成度を上げようということでしたね。

――MSXPLAYerのできは、山下さんから見てどうですか。

山下 感心するばかりで、FM音源はさすがに動かないだろうと思ったら、ちゃんと実現されたりとか、ある意味びっくりしましたね。

――「ここは大丈夫だろうか」みたいなところはありましたか。

山下 すごく心配だったのは、インテントが足を引っ張っていないかという点です。裸のOSの上ではサクサク動くものが、インテントの上に載っているために使えるハードが限られるのではないかというのが一番心配でした。おかげさまで、ハードウェアの性能が上がってきたこともあって、動くマシンが何種類かあってよかったなと思ってます。もう少し動作が速くてもいいんじゃないのとかと思う人もいるかもしれないのですが、それについては、もう少しチューニングできたらいいなと考えています。

――チューニングの可能性はあるわけですか。

山下 もちろんあります。

ただ、プロファイラがあればどこでどのくらい時間がかかっているか特定して、たくさん食ってるコードからチューニングしたりアセンブラで書いてスピード上げることができるので楽なのですが、インテントではまだそうしたツールの調達がしんどい。MSXPLAYerをプロモートする立場からすると、MSXPLAYer自体インテントの世界ではトップランナーなんです。他に、エンターテインメント系のアプリケーションとしてま

とまったものはありません。だから、逆にMSXPLAYerに影響されて、そこから誰も経験していない世界のパイオニアが生まれ、どんどん開拓していってくれることを望んでいるんです。最近、京セラさんからインテント搭載の「Pocket Cosmo」というPDAが発売されて、いろいろなところで使われ始めていますので、ツール類も整備されるといいなと。こう言っていると、いずれ動いてくれる人も増えてくるだろうということで(笑)。

――なるほど。ところでMSXPLAYerは、MSXの仕様をほぼ満たしているわけですから、Windowsを完全に隠して、MSX専用マシンとして使うというのはどうでしょう。

山下 問題はスロットですかね。USB接続の外部スロットなんかを誰かが作れば、昔のROMカートリッジも使えますし。

――それは夢ですね～(笑)。

MSXの仕様を守ること

――Mマガ版MSXPLAYerに、スピードのリミッターを解除する∞ボタンをつけてほしいとMSXアソシエーションに提案したんです。今MSX速度を体感してみると、起動や、BASICのリスト表示などがひたすら遅くて耐え難かったので。そうしたら「うーん、それはMSXの仕様じゃないので」と考え込まれまして、そういうものなのかと思いました。

山下 MSXのリバイバルだという立場からすると、いかに忠実に、それこそバグまで再現するんだという人もいますからね。

――なんとか、そちらのボタンは実現してもらって重宝してますが、そのついでに「スプライトが一定数以上横並びすると表示されないという制限をなくすボタンはどうだろう」なんてことも言ってみたんですが、そちらについてはさすがにボツでした(笑)。

山下 互換性があるからこそMSXたりえたわけで、まずスタンダードを守らないとMSXでなくなってしまう。MSXであればこその本質的なことかもしれませんね。ただ、MSXがソフトウェアになっちゃったわけです。ハードウェアの場合には仕様を変更すると互換性が失われます。しかしソフトウェアならエミュレータごと配ることもできる。「このアプリケーションを動かすための専用エミュレータです」というのも全然おかしくありません。仕事に使う機械じゃないので、そんなに堅く考えなくてもいいんじゃないかとも思いますが。

――ならば、もう一歩つっこんで、MSX開発の第一人者である山下さんとしては、MSXPLAYerをこう発展させていきたいという思いはありますか。ぶっちゃけて言えばMSXPLAYerによるMSX3とか。

山下 僕としては、実はMSX2＋ぐらいがちょうどいいのではないかと思ってるくらいなんです。あれ以上重くしても、それでおもしろさが増すかと言えばそうはならない。たとえば、スプライト無限に使っていいよとなったとき、それでおもしろくなるのかと。プログラムコードを書くのは楽かもしれないですが。

僕がMSXに関わっていた当時、IBM PC互換機を中心にどんどん技術が進歩していましたから、みんなそれを追っかけてたんです。もともと私はハードウェア屋的なところがありますし、メーカーさんとつき合っていると「時代遅れのハードじゃ売れるわけないよ」となります。そして、どう追っかけようかとキャッチアップすることばかり考えていました。結果的にMSXは続かなくなったんですが、果たして続けていたらよかったかというと、多分IBM PC的なものを作っていたわけですよ。すると、ものすごいソフトウェアの塊になって、もはやMSXのおもしろさがどこにもないものになっていたと思います。あそこあたりで止まって、ある程度コンパクトな、1人の人が100%理解できる規模が理想ですね。

――それはとてもわかる気がします。今はそういうPCが少ないですから。

山下 MSXも最後はちょっと重くしすぎちゃったかもしれません。だけど、最初の頃はメモリがほんの数十Kバイト程度でしたよね。それが最後はMバイト単位で積んでいますから、そうした経緯の中では、シングルアーキテクチャの中で、まあまあうまく作れたかなとは思いますが。

いつでもどこでもが僕のMSX3

――やっぱり、MSX3はありませんか。

山下 MSXが目指していたのは"アプリケーションプラットフォーム"なんです。当時は、MSXのソフトウェアとハードウェアをもっと重くしていく技術的志向性が強かったんですけども、本質的に重要なのはそこではありません。「このプラットフォーム用にアプリケーションを作っておけばどこでも動くよ」ということなんです。実際には、MSXというハードウェアは生き続けられなくなってしまいました。しかし、こうしてソフトウェアになったことでゾンビのごとく生き返った(笑)。ソフトウェアですから身軽です。ハードウェアは売るのが大変ですが、ソフトウェアなら自由自在に広めることができます。FDDにはベルトが弛むなど命に限りがありますが、ソフトウェアならいつまでも生き続けられます。できればどんなOS、CPUでも動く。これこそMSXで実現したかったことです。

インテントの目指しているのは、まさにそれなんですよ。OSやCPUに依存しない"アプリケーションプラットフォーム"です。そうした観点からすると、こいつは僕にとってのMSX3じゃないのかと。もう1回やりたかったのは実はこれなんだ。ちゃんとできてるじゃないかと。環境に依存せず、どこでも使えるのがMSX3なんです。どんなマシンでもインテントを積むと、それはMSX3です。京セラのPDAもMSX3です。何もかもが

MSX3になって、1つソフトを書いたらどこでも動くという世界です。

一方、MSXPLAYerの立場からすると、インテントの上にMSX PLAYerが載っているということは、インテントが載る機械はすべてMSXになります。だからMSXは、どこででも動くユビキタスコンピュータになるわけですね。僕としては、そんな絵が気に入ってるんですけど(笑)。

――考えてみると、どんなCPUやOSでも動作するというのは、それこそMSX3と呼んでもいいほどの質的変化ですね。

山下　結局ハードウェアというのは、アプリケーションがあってナンボのものなんですよ。アプリケーションにエネルギーを投じた人の文化資産ですよね。もっと大げさに言っちゃうと「人類の文化遺産」ですよね。腐らせることなく、未来永劫どこでも使える必要があるでしょう。はるか昔に書かれた文学作品も、ちゃんと価値を持っているわけです。そういう意味で十数年前、誰かが知恵を絞って作ったソフトウェアがどこでもいつまでも動くというのは意味のあることだと思います。

マーケットの可能性

――こうしてMSXが蘇ると、関係する仕事が増えるかもしれませんよ。

山下　そうなんです。私もセガで1年以上仕事しているんですが、ゲームって1つの大きなアプリケーションですよね。CPUとメモリがついてる装置でなにをするのかというと、必ず何%かの人がゲームをやるわけですよね。ならば、そのゲームをどうやって作るかというと、やっぱりお金がかかります。今はXBoxのような高性能のマシンがありますが、一方で携帯電話やPDAなどの機械を手に取ると、ちょっと前のゲーム機ぐらいの性能です。そうなると、できることはわかってしまっていると言えます。昔も今も内容はそんなには変わりません。当時の人が一生懸命考えて、その機械でできることをギリギリまで引き出していたのだから、今になって急に新しいアイデアがひらめくわけでもない。そういう意味で携帯電話やPDAが出てきても、その環境でまったく新しいゲームを開発しようというのはあまり賢くなくて、かなりの部分は過去作ったものと同じでいいんじゃないかなと。

――最近、過去のゲームをもう一度やりたいという人が増えているようですよ。

山下　たとえばセガでいうと「ぷよぷよ」ですか。あれは今でも十分おもしろいわけですよ。何年か前にコンパイルさんが、最初はMSXで作られて、その当時のハードウェアでああいうおもしろいものができたわけですが、携帯電話で今ぷよぷよをやっても同じようにおもしろい。だから、昔のゲームが動けば、かなりのニーズをそこで満たすことができる。そういう意味で、エミュレーション技術で過去のソフトウェアを蘇らせるというのはソフトウェア業界からみてもよい方向だと思います。

ハードウェアとしてはPocket PCとかいろいろなものが出てきていますが、ゲーム作りますか？と言われるとなかなか誰も作らないわけですよ。セガでも、台数何台で、いくらの値段つけて……と考えて「うーんそれは」となっちゃいます。そこに過去のゲーム資産を持ってくることができれば、マーケットとして十分大きくなっていなくてもちゃんとゲームが供給できる。それは誰にとってもいい話です。お客さんは過去のおもしろいゲームで遊べる。ソフトウェアハウスは過去に作っていたソフトで、たくさん儲からないにせよビジネスとしての可能性が出てくる。新しいプラットフォームを立ち上げるときの呼び水的な供給ができるでしょう。セガも、MSXのゲームを20本近く出しているんで、こういうものを手がかりに考えていきたいですね。

――それはとても楽しみです。

山下　ソフトウェアって、さっきの大上段振りかぶった言い方をすると"文化資産"じゃないですか。それは使われて初めて価値が出るものです。皆が使うためにはどうしたらいいのかというと、今度セガも始めたんですが「どんどんコピーしてください」とマニュアルの最後に書いてあるんです。だけど、お金払ってもらわないとこちらは潰れちゃうんで、コピーしてもらったものは、初めはお試し版として動いて、"鍵"を買ってもらうと正式版になるという仕掛けです。ソフトウェアが人から人へ広まるというのは自然な流れです。ネットワークもどんどん速くなるし、簡単に手に入ります。だからこそ「気に入ったらお金を払う」というシステムをちゃんとやりたい。それができないと良いアプリケーションプラットフォームはできない。そういうこともMSXPLAYerでお手伝いできたらなと思ってます。

――今日はどうもありがとうございました。

PROFILE

山下良蔵。1953年広島県生まれ。大阪大学卒業後、福島県立医科大学でマイコンにを使った生理学の研究をする。その後マイクロソフト社で最初の日本語版Microsoft Basicを手がけ、さらにアスキー・マイクロソフトFE本部でMSX～MSX turboRに至るまで開発をしてきたMSXの第一人者。今年、セガへの転籍となり忙しい毎日だが、MSXPLAYerにも相談役として関わっている。

MSXPLAYerを支える技術 intentとは?

●TEXT/編集部
取材協力/タオ・ジャパン株式会社

MSXPLAYerを支えるintent

MSXPLAYerは楽しんでもらえただろうか。MSXPLAYerは、現在、Windows版とPocketPC版がリリースされている。本誌収録こそ間に合わなかったものの、Linux版がすでに準備中であり、Macintosh版も検討中とのことである。昨年末、「MSX 電遊ランド」での制作発表から約1年。短期間のうちに多くのプラットフォームに対応していることがわかる。

実は、MSXPLAYerの素早いマルチプラットフォーム対応には、英Tao Group社が開発したintentという技術が使われているのだ。

バイナリとプラットフォーム

intentを一言で言えば、「バイナリ互換の共通プラットフォーム」となる。これだけでは意味がわからないかもしれない。ここで、ちょっとだけバイナリとプラットフォームについておさらいしておこう。

コンピュータのプログラムは、実行時点では、マシン語のバイナリコードになっていなければならない。バイナリコードは、CPUが直接解釈可能な命令やデータで構成されるが、その命令はCPUの種類によって異なる(互換CPUなどの例外はある)。そのため、バイナリコードで記述されたプログラムを、異なるCPUを搭載するコンピュータでそのまま実行することはできない。

また、CPUやハードウェアの構成がまったく同じでも、OS(オペレーティングシステム)が提供する機能を利用しているプログラムは、別のOS上でそのまま動かすことはできない。これは、たとえば、同じPC互換機上で動くWindowsとLinuxにバイナリレベルで互換性がないことからもわかる。

同じバイナリを、異なるプラットフォームで実行させるには、特別な手段が必要なのである。

エミュレータという考え方

その特別な手段の1つが、ソフトウェアエミュレータだ。読者は、すでにMSXPLAYerを使って、Windows上でMSXのバイナリコードを実行しているから、この方法は理解しやすいだろう。PentiumやAthronなど、x86互換CPUを搭載するWindowsパソコンの上で、Z80CPUを搭載するMSX用のプログラムが実行可能になっている。

この仕組みを一段掘下げて考えてみよう。MSXPLAYer自体を、異なるプラットフォームで実行可能にするにはどうすればいいだろうか。仮にMSXPLAYerがWindowsのバイナリコードでしか存在しなければ、Pocket PC 2002やLinuxに、Windowsエミュレータを用意すればいいはずだ。

しかし、Linuxならまだしも、Pocket PC 2002は、Windowsパソコンに比べて、CPUの処理能力やメモリ容量が劣るPDA上で動いている。その上で、自身より巨大なWindowsをエミュレーションするなど、至難のワザといわざるを得ない。さらに、プラットフォームの種類で言えば、携帯電話や自動車、家電製品、工場の機械など、我々の身の周りには、パソコン以外のコンピュータも数多く存在する。これらに使われる、いわゆる組込み型のコンピュータについては、PDAよりもさらに大きな制約を受けることになる。

そこで、Windowsのような巨大なものではなく、応用範囲の広い、もっとコンパクトで軽快なシステムを考案し、そのシステムを色々なプラットフォームでエミュレーションさせようというのがintentである。

intent RTOS

intentで提供される環境の核となるのは、32ビット仮想CPUであるVPと、専用のマルチタスクリアルタイムOSを一体化させたシステム、intent RTOS(以降RTOS)だ。

RTOSの仮想CPUは、ハードウェアとして実在はしないが、その仕様は、32ビットRISCアーキテクチャとなっている。架空のCPUとは言え、きちんと設計されているので、アセンブラでプログラムを記述することも可能だ。

RTOSは、WindowsやPocket PC 2002などのホストOS上で動作し、ホストOSと共存することが可能になっている(図)。

intentが対応するCPUを搭載するハードウェアならば、RTOSを直に実装することもできる。機械制御のようなリアルタイム性が要求される用途では、このほうがRTOSのリアルタイムOSとしての真価が発揮されるのだ。

RTOSが対応するCPUと、共存可能なホストOSを表に示す。

表を見ればわかるように、RTOSをターゲットに、ひとたびバイナリコードを作成しておけば、同じものを、WindowsやLinux、Palm OS上でも実行させることができるのだ。

Javaとの違い

ここまでの説明で、Javaの実行環境を思い浮かべた人もいるのではないだろうか。Javaも、Java仮想マシン（JVM）を実装しておき、その仮想マシンをターゲットにコンパイルしたバイナリコード（バイトコードと呼ぶ）を実行するという点で似ている。

Javaとintentの決定的な違いは、開発言語にある。Javaの開発言語はJavaしかない。しかし、intentの開発言語は、アセンブラ、C言語、C++を始め、プログラミングに必要とされる言語のほとんどを網羅している。もちろんその中の1つの選択肢として、Javaも選べる。

たとえば、プログラミングの現場では、C言語で全体のプログラムを記述しておき、ハードウェアと同期をとったり、速度が要求される部分だけをアセンブラで記述しておくという手法がとられる。こうした手法が、intentならそのまま行えるというわけだ。これはJavaにはできない手法である。

intent JTE

とは言え、現在の状況では、Javaは無視できないものとなっている。そこで、intentには、Java実行環境として、intent JTE（Java Technology Edition）が用意されている。JTEは、Personal JavaとKVM（K Virtual Machine）で構成される。

JTEは、Javaプログラムのバイトコードを、実行前にRTOSのCPUであるVPのネイティブコードに変換して実行する。この変換（トランスレーションと呼ぶ）が非常に高速で、JIT（Just In Time）コンパイラ相当の実行速度を実現するという。

実際、今年9月に発売された京セラのPDA「Pocket COSMO」は、Strong ARMプロセッサにintentをネイティブ実装したJavaマシンに仕上がっているが、Javaで作られたプログラムがとても軽快に動く。

また、多くのJVMでは、ガベージコレクション（以降GC）と呼ばれる処理が自動的に行われる。これは、プログラムが利用しているメモリ領域内の未使用部分を回収して再利用するための処理だが、ひとたびGCが開始されると、プログラムの実行が遅くなったり、処理が断続的になることもある。それは仕方がないが、開始のタイミングを予期できないというのが困ってしまうのだ。

たとえば、機械制御やゲームへの応用を考えるとしよう。機械制御は、タイミングが命といってもよく、常にプログラムと制御すべき機械の同期をとっておく必要がある。GCが開始されて処理が遅れると、機械が暴走しかねない。また、アクションゲームで、突然画面の書き換えが遅くなったり、止まったりしては、ゲームとしてはとても成り立たない。その点、JTEでは、GCの実行をプログラマがコントロールすることも可能になっているので、こうした事態は避けられる。

ぐっと身近になるintent

intentは、先に紹介したPocket COSMO以外にも、すでに有名メーカーのAV機器などに組み込まれ、我々の身近なところで活躍している。そして、注目されるのは、なんと言っても次世代携帯電話だ。高速でコンパクトなJava実行環境として、intentは携帯電話にもってこいの技術なのだ。

聞くところによれば、来年後半あたりからintentを採用した携帯電話が登場するというので、期待しよう。

intentは、今後も目が離せない技術である。

表　Intent対応CPUと共存可能なホストOS

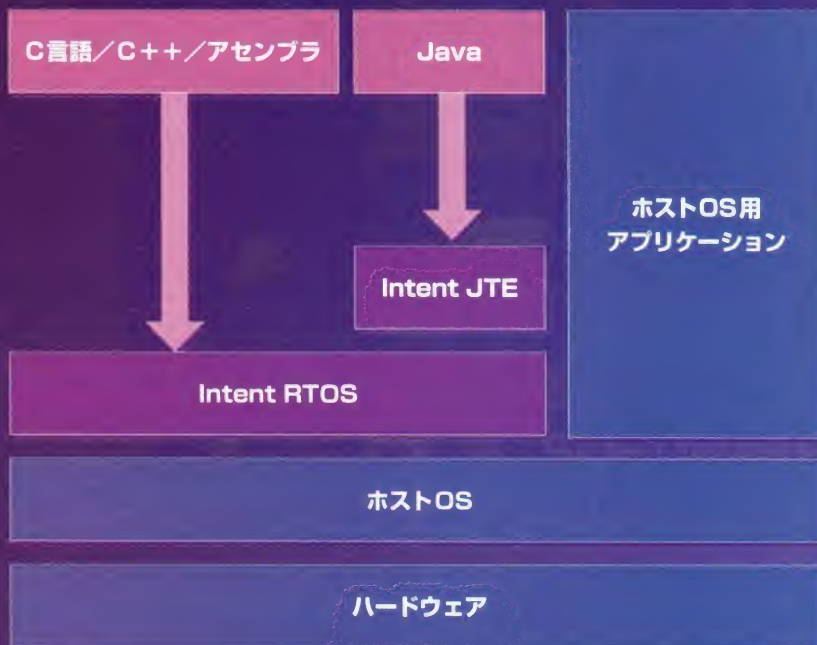
●対応CPU

- Intel x86互換、Pentium互換
- Intel ARM/Strong ARM/Xscale/OMAP
- Motorola/IBM PowerPC
- NEC V850
- 日立 SH3、SH4
- Motorola Mcore、Cold Fire
- MIPS
- Patriot Scientific PCS1000

●対応OS

- Linux/Embedded Linux
- Windows 95/98/Me/NT/2000/XP
- Windows CE/Pocket PC
- MS-DOS
- OS9
- QNX
- VxWorks
- ITRON
- EPOC32
- Palm OS

図　IntentとホストOSの関係

関連URL

英Tao Group社　http://tao-group.com/
タオ・ジャパン株式会社　http://tao-group.jp/
Intentファンサイト「intent FAN」　http://www.intent-fan.com/

本誌に収録したMSXPLAYerは、MSX史上初の「公式エミュレータ」だ。
何をもって“公式”なのかといえば、MSXの権利を管理しているMSXアソシエーションが承認している点にある。
では、MSX公式エミュレータのプロジェクトはどのように生まれ、そしてそれがどう展開し、
将来的な展望はどうなっているのだろうか。
このあたりを編集部が関係者に直接取材してまとめてみたので、MSXユーザー的には、ぜひ押さえておいてほしい。

プロジェクト MSXPLAYer

TEXT：編集部　協力：MSXアソシエーション

同人サークルがMSXを強く牽引

まずは、歴史的経緯を述べておこう。次ページの年表も合わせてご覧いただきたい。

1983年にMSXは発表され、1992年に新機種の開発を中断することになるが、MSXユーザーは弊社MSXマガジンがその年の夏号以降書店に並ばなくなってしまったことで大きな失望感を覚えることとなる（※1）。それでもMSXから離れたくなかった多くのユーザーは、アスキーネットやTHE LINKSなどのパソコン通信や雑誌MSX・FANに心の拠りどころを求めた。

だが1994年には、MSX実機もPanasonic FS-A1GTを最後に生産中止となり、さらに、MSX・FANも1995年８月号を持って休刊となる。そんななか、同誌は休刊前の約１年をかけ「MSXを同人活動で支えよう」と呼びかけ、それに呼応するように全国に多数のサークルが作られていった。

1995年に行われた「MSXパーティ」は、ユーザー主導によるはじめての大型MSXイベントであり、メーカーが撤退したら諦めるという常識に反旗を翻した転機となる。同時期「MSXキャラバン」が主要８都市で行われ、全国に散っているユーザーが徐々に連絡網を整備、組織化しゲーム等のソフトや同人誌を発表していく。そうしたイベントにおいて、当時入手困難だった海外製エミュレータ「fMSX」がユーザーの目に留まる。しかし、他のPC上でMSXが動作することは驚きではあったが、この時点でエミュレータという仕組みにMSXの未来を見る者はほとんどいなかった。むしろ、本体を修理して延命させたり周辺機器を自作しながら、将来MSX本体を自力で再生産することを夢みていたのだ。

ユーザー達は仲間を求め、作品の即売会や交流会を盛んに開催した。1996年から1997年にかけては、こういったイベントが、全国で年60回以上行われるまでに至っていた。価格も安く、ユーザー数も多かったMSXは、メーカーの撤退を乗り越えハードとソフトの両面から支えあえたのだ。これは、結果としてMSXが、ゲーム、CG、作曲、あるいはハードの自作といったクリエイティブな道具として魅力的だった証といえる。表現力では他の高額なパソコンやゲーム機と比べると劣る面が多いが、そういった制約があるからこそMSXはおもしろいのだ。

内部情報の公開は積極的に行われていたがユーザーはより深く解析し開発ツールを作り、名作ゲームを生んでいく。本誌掲載のインディーズゲームも、こうして誕生した。市販ソフトにも負けない出来栄えといえるだろう。

1997年11月に開催された「MSX World Expo」の成功によりユーザーは継続的な活動に自信を深め、新規ユーザー獲得のためイベントの大型化を模索しはじめる。しかし、大型化は逆に人手不足や開催費用など運営面で新たな課題を生んだ。そしてインターネットが一般に広がりはじめる1998年以降は、ユーザーの活動がインターネットに移行し、同人ソフト入手の意味ではイベントの重要性が低くなっていく。ユーザーは徐々に高齢化し、ハードウェアも老朽化し、漫然とした閉塞感が漂いはじめた。MSXはどうしたら生き残れるのか、今後の戦略をトータルに検討せねばならない時期にさしかかっていたのだ。

1999年初頭に、何名かのユーザーがアスキーの西 和彦に面会を求めた。それまでも西和彦に何らかのアピールを行おうという提案は散発的にあったが、実際に行動に出たのはこのときが初めてであった。そしてユーザー自身の手で大幅に機能を拡張したMSXシステムを西に見せたのだ。西はこの時はじめて同人ハードとソフト、その活動状況を知るところとなる。そこにはMSXマガジン92年夏号で西が語っていた次世代MSXが実現されていた。CD-ROMやHDDをMSXから動作させ、動画を再生し、インターネットに接続してみせた。西は驚き、そして大いに喜んだと言う。これが大きな転機となった。

※1　MSXが盛んだった中盤から終盤にかけては雑誌「MSXマガジン」「MSX・FAN」「マイコンBASICマガジン」などがユーザーをひっぱっていた。また、コンパイルは独自のアイディアであるDISKマガジン「ディスクステーション」を発表し人気を博していた。また、TAKERUというソフト自動販売機によって数々の市販ゲームや、同人ソフトが供給されていた。

年	月	MSXにまつわる出来事
1983年	6月	MSXの仕様が発表される
	10月	「MSXマガジン」0号発売
	11月	「MSXマガジン」創刊号を発売
1984年	1月	各社のMSXマシンが出そろう
1985年	5月	MSX2の仕様が発表される
	10月	出荷台数100万台を突破
1986年	12月	MSXネットの運営開始
1987年	11月	ディスク内蔵低価格機の登場
1988年	9月	MSX2+の仕様が発表される
1989年	7月	ハードディスク(SASI)インターフェース発売開始
1990年	1月	MSX出荷台数が国内外で400万台を突破
	9月	MSX turboRの仕様が発表される
1992年	8月	「MSXマガジン」92年夏号を発売
1994年		Panasonic FS-A1GTを最後にMSXの実機販売が終了
	7月	「MSX·FAN」1年後の休刊を予告
1995年	3月、8月	ユーザー主導の大型イベントとして「MSXパーティー」(SYNTAX)開催
	7月	「MSX·FAN」(徳間書店)休刊。同誌において、西 和彦が「ハードウェアスペックとソフトウェアで定義するイレートテクノロジー」について発言。北根編集長、同誌においてMSXユーザーにサバイバルを提言
	7月、8月	主要8都市にて「MSXキャラバン」(GENUINE NETWORK)開催
1996年〜		同人イベントが盛んになる。年60回以上が開催された
1996年	8月	「MSXフェスタ 96」(モオソフト)開催
1997年	8月	「MSXフェスタ 97」(モオソフト)開催
	11月	「MSX World Expo 97」開催
1998年	8月	「MSXフェスタ 98」(主催:モオソフト)開催
1997年	1月	MSXのソフトも販売していた自動ソフト販売機「TAKERU」が店頭から姿を消す
1998年〜		同人サークル活動のインターネットへの移行が進行
1999年	1月	西 和彦(当時アスキー取締役)が初めてユーザーグループと会見
	8月	「MSX電遊ランド」開催。西 和彦が公式エミュレータへの考えをコメント
	11月	「MSX World Expo 99」開催。「MSXリバイバル宣言(サバイバルからリバイバルへ)」
2000年	8月	「MSX電遊ランド2000」開催。西 和彦が公式エミュレータの開発を表明
	9月	西 和彦がMSX復活計画を「MSXリバイバルプロジェクト」と命名
	10月	公式エミュレータの開発スタート
	12月	「第18回 Salon del MSX」(スペインのバルセロナで開催)にて、MSX公式エミュレータα版を初めて公開
2001年	5月	「MSXティルブルグフェア」(オランダのティルブルグで開催)にて、西 和彦がMSXPLAYerの開発の進捗状況を発表
	6月	リバイバルプロジェクトが公式エミュレータの名称を募集
	8月	西 和彦がMSXPLAYerロゴのデザインを作成
	9月	「MSX電遊ランド2001」開催。MSX公式エミュレータβ版のテスト配布
	10月	「MSXアソシエーション」の名称が初めて公表される
2002年	2月	「MSXアソシエーション」設立
	12月	「MSX MAGAZINE 永久保存版」発売。MSXマガジン版MSXPLAYerを付属CD-ROMに収録し、一般公開頒布
2003年	6月	MSX規格の仕様発表から、20周年を迎える。MSXをテーマにした大型イベントを企画中

「MSX World Expo」の会場風景

ユーザーグループにより拡張されたMSXハードウェア

電遊ランド2000のポスター

電遊ランド2001での西 和彦氏の講演

プロジェクト
MSXPLAYer

パソコンの性能向上によりエミュレータが登場

一方で世の中では16ビットパソコンから32ビットパソコンへの移行が進んでいた。低価格のPC/AT互換機がちまたにあふれだし、CPU等の高性能化によってパフォーマンスは加速度的に向上していった。

そんな状況のなか、MSXをエミュレータで再現しようという発想が生まれたのは自然なことだった(※2)。今でこそ、ありとあらゆるコンピュータのエミュレータが作られているが、MSXエミュレータが登場したのはかなり初期である。確かにエミュレータのようなデジタル的クローン技術にはロマンはあるだろう。もっと単純に昔のゲームをしたい、という欲求もわかる。しかし、MSXエミュレータが早くも世に現れた理由は、やはりMSXの内部情報が積極的に公開されていたからではないだろうか。

しかし、MSXを完全にエミュレートすることは簡単ではない。CPU、VDP (Video Display Processor)、ディスク、その他もろもろの環境を再現し、相互にタイミングをとってデータアクセスをしなければならないのだ。したがって、ゲーム等が動作しない、挙動がおかしいことは当たり前。また、インストール方法や各種設定方法が複雑でわかりにくいなど、誰でも気軽に使えるものではなかった。さらに一番の問題は著作権。エミュレータにはBIOSやBASICのROMイメージが必要だ。

お手軽かつ安心して使える公式のエミュレータを望む声が、ユーザーの間で、徐々に出始めていたのも当然と言えるだろう。

※2 エミュレータという発想そのものは、コンピュータの世界では珍しいものではない。ただ、かつてはパソコンで他のコンピュータをエミュレートするのは不可能に近かった。一般的に、あるコンピュータで別のアーキテクチャのコンピュータを再現するには、桁違いのパフォーマンスを必要する。ちなみに、MSXで採用していたZ80 CPUの動作クロックは3.58MHz。現在のパソコンは3GHz、比べれば数百倍のパフォーマンスを有している。しかしMSXはCPUだけではない。今日の高性能PCのスペックをもって、ようやくMSX全体を再現できるようになったのだ。

※3 その後「MSX電遊ランド」では、MSXゆかりの人物が講演を行うのが恒例になっている。アスキーでMSXを開発した技術者である山下良蔵(P72参照)、松下でMSX普及に尽力した前田一秦が、またゲストには元MSX・FAN編集長の北根紀子や株式会社コンパイル代表取締役の仁井谷正充も登場した。

インテントというプラットフォームの戦略に合致

MSXといえば西 和彦。1978年に株式会社アスキーを設立し、1979年には、まだほんの小さな会社だったマイクロソフトのMicrosoft BASICを日本の各メーカーに売り込み、MS-DOSなどの日本語化や販売を行うなど、いわば日本にパソコン文化を広めた人物である。そして世界統一規格ホームコンピュータである「MSX規格」の提唱者であることは言うまでもない。

その西 和彦は、アスキーに在籍中、英国Tao社の「イレート」(インテント)というマルチプラットフォーム環境を見い出し注目していた。これは、どんなCPU、どんなOS上でも同じプログラムが動作する環境を提供する技術だ(p.76参照)。MSXが世界共通規格を目指していたというなら、イレートは世界共通プラットフォームを目指すもの。そう考えると同じ方向性を感じずにはいられない。

1999年初頭にMSXユーザーグループから、毎年夏にユーザーが開催しているMSXイベントでの基調講演を依頼された西は、秋葉原のメインストリートでのイベント開催を提案。1999年8月に、西参加のもと「MSX電遊ランド」が行われた。そのとき会場に訪れた西は、会場に入りきれないほど集まったユーザーの熱気とその活気に改めて感銘を受けたという。ユーザーは西が、西はユーザーがMSXを見捨てていなかったことに感動した。そして「ホームPCとしてのMSXの16年を総括し21世紀のPC戦略を提案する」と題した講演(※3)にて、会場からの質疑応答に対し、約束はできないとしながらも「公式のエミュレータも考えなければならないと思っている」と語り、1年後の「電遊ランド2000」では、インテント上で動作する公式エミュレータの開発をファンの前で正式に表明するに至るのだ。

MSX公式エミュレータの開発プロジェクトがスタート

こうして2000年10月から公式エミュレータの開発がスタートを切る。ここには多くのユーザーも協力することになる。公式エミュレータを作るにあたり期間を短縮するため、既に存在していたエミュレータ(fMSX)を正式に承諾を得て、このコアを利用して開発が進められた。従って開発者の最初にして最大の課題は、インテントというまったく新しいプラットフォームへの挑戦だった。なお、インテント上ではさまざまな言語が使えるが公式エミュレータはC言語で書かれている。開発環境はWindowsだ。国内外の有志が、VDP(Video Display Processor)や音源まわりについても、合わせて開発を進めていった。

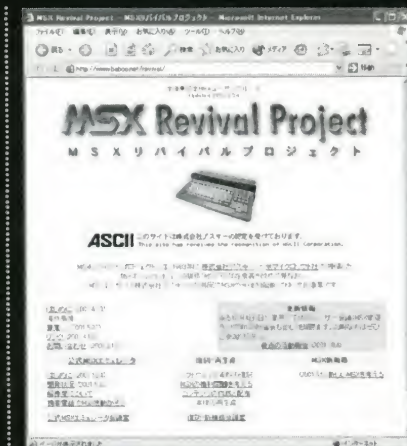
開発には、実は数社のメーカー技術者も携っている。さらに、動作をより実機に近づけるため、多数のテスターの手も借りている。たとえば、あるゲームをテストプレイ中にMSX実機と音色が異なることがわかり、数々のゲームで同様の現象が起こるかを調査し、最終的には多くのユーザーの情報提供と協力によって原因が解明され、MSX実機の音源とほぼ同じに聞こえるように調整されたという。画像や動作速度、操作フィーリングについても、同様なフィードバックがあった。このように多くの人の協力があってこそ、より完成されたものにすることができたわけだ。

MSX公式エミュレータの名前が一般公募された

MSX公式エミュレータの開発がスタートすると同時に、有志によって「MSXリバイバルプロジェクト」(※4)というサイトを公開。ユーザーからの要望を拾い上げる場とした。具体的には、開発状況の公開、「解像度について」と題する画面のアスペクト比についての考察、権利問題についての意識改革、会議室の提供などで構成されている。

そんななかで、もっとも話題を集めたコンテンツが「公式エミュレータの名称を募集」という一般公募。採用されたとしても、賞金もなにも出ないにも関わらず、多数の応募が寄せられた。

そして西 和彦が公募作のなかから選んだのが「MSX-PLAYER」だ。その理由として、MSX-PLAYERに2つの意味合いを感じたからだという。1つは「MSXのソフトを再生する環境という意味でのプレイヤー」、もうひとつは「MSXで遊ぶ(使いこなす)人々という意味でのプレイヤー」である。これは「MSXのコンテンツをいつでも、いつまでも使えるようにしたい」、そして「MSXでパソコンの勉強をしたり、使いこなそうという人に環境を提供し続けたい」という願い、MSXとMSXユーザーへの思い入れの表れだと言えるだろう。

そして紆余曲折を経て、「-」が削られ、2つのプレイヤーの意味合いを強調する意味で「ER」が小文字になり、最終的には「MSX PLAYer」という名称に確定した。

リバイバルプロジェクトのホームページ

※4 「MSXリバイバルプロジェクト」のサイトでは、MSXユーザーグループの究極的な目標が「MSXユーザー自身の手でMSXの文化を守り通せる環境を作ること」とされている。また、具体的には「①手軽に作って遊ぶ行為の意義を伝えてゆくこと」「②MSXに関する資産(ソフトウェアや資料など)を残すこと」「③MSX実機を自作できるようにすること」とある。そしてとにかく、できるだけ自分たちでMSXを推進していくのだという決心が書かれている。なお、現在は人手不足で更新はストップしたままだ。

名称	読み方	語源
Advanced MSX	あどばんすど えむえすえっくす	
Amee	えいみー	Ascii Msx Emulation Environment
eMSX	いー えむえすえっくす	Elate OS, Emulator
eMuSX	えみゅえすえっくす	Emulator + MSX
INFINITY	いんふぃにてぃー	MSXの生命に限界はない
Live MSX	らいぶ えむえすえっくす	
MAPS	まっぷす	MSX Advanced Plugin System
Mesa	めさ	Msx Emulation System Apprication
MeMSX	みー えむえすえっくす	The Millennium emulator for MSX
MIND	まいんど	Msx IN Diffrent-device
MSX3	えむえすえっくす すりー	
MSX-A	えむえすえっくす えー	Advance、Anywhere、Any、ASCII
MSX ACE	えむえすえっくす えーす	Ascii Corporation Emulated
MSX Alive	えむえすえっくす あらいぶ	
MSX Egg	えむえすえっくす えっぐ	この卵をみんなの力で育てていこう
MSX-Me	えむえすえっくす みー	私のMSX、2000年に企画立ち上げ
MSXNG player	えむえすえっくす えぬじー ぷれいやー	stand for MSX Next Generation player (MSX次世代プレイヤー)
MSX-PLAYER	えむえすえっくす ぷれいやー	MSXコンテンツを動作させるもの
MSX replayer	えむえすえっくす りぷれいやー	
MSX Standard	えむえすえっくす すたんだーど	「標準」ではなく「規格」という意味で
MSX V.T. Player	えむえすえっくす ぶいてぃー ぷれいやー	MSX Vintage Technology Player
MSX ZERO	えむえすえっくす ぜろ	復活ではない、ここからが始まりの意
PlayMSX	ぷれい えむえすえっくす	MSXで遊ぼう!?
MSXV	えむえすえっくす ぶい	アスキー社内でとりあえずついた名称。Virtual Machine Super X
XEVES	ぜうぇす	msX Emulation.Virtual Entertament System

※主な応募名称のみを掲載

<MSX公式エミュレータの名称募集のルール>

1. できるだけシンプルであること
2. MSX実機の入力で表記できること
3. 海外ユーザーも念頭に国内外で利用できる文字であること(全角文字、グラフィック記号、かな文字の使用は禁止)
4. 雑誌やカタログに掲載される(かもしれない)ことを想定して語感や見栄えを考える
5. 商標登録済や一般的に普及している語句でないこと
6. その他、公序良俗に反しない語句であること

MSXアソシエーションの設立

開発がスタートしてから1年あまりが経過し、ようやくMSX公式エミュレータが実用に耐えるものとなってきた。実行環境が整えばゲームの復刻や保存、技術資料の発掘、整理などのために各権利者との交渉がスタートできる(実行環境がなければ交渉もできない)。そうなると、こちらも対外的な告知や折衝のために団体が必要となる。そこで「MSXアソシエーション」が設立されることになった。代表はもちろん西 和彦。そのほか開発に携わったメーカー各社より理事が選出されている。

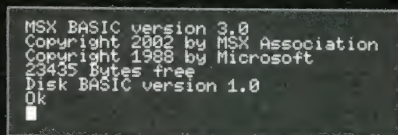
MSXアソシエーションのメンバーの多数は専属ではなく、ボランティアスタッフだ。したがって、二足のわらじを履きつつ活動を行っている。彼らの主な仕事は、本誌のような媒体への企画協力といった活動、ゲームやシステムの権利者との交渉、各種資料やソフトウェアの発掘、さらには現在もっとも重要なMSXPLAYerそのもののテストだ(※5)。

※5 MSXPLAYer β版は、現在よりもかなり動作が不安定で、「ザ・キャッスル」もまともにプレイできないぐらいだった。また、ディスク版のゲームを再生する機能もなく、音も鳴らなかった。なんとかなってきたのはここ半年ぐらいだ。もちろん本書付属の正式版は、β版に比べ格段に完成度が高まっている。

ずいぶん完成されてきているとはいえ、MSXPLAYerは今もって発展途上である。したがって、新しいバージョンのMSXPLAYerがリリースされるたびに、多くのプログラムを動かし、その挙動をくまなくチェックしなければならない。徹底した品質管理のためにはテストプレイが何より重要となるのだ。当たり前のことと言えばそれまでだが、手弁当のスタッフだけでそこまで徹底することは簡単なことではない。

特に、正式公開するようなゲームの場合には、必ず1本1本最後までクリアし、問題なく遊べることを確認しているという。むろんクリアするだけでなく、バグが出そうなところを重点的にチェックしなければならないわけで、ただゲームが好き、というだけではこなせない作業といえる。このように、徹底した品質管理によってMSXPLAYerの信頼性は日々高められているのだ。

ちなみにスタッフは、MSXがピークだった1989年ごろ中高生だったMSXユーザーが多いそうだ。いまでは、メーカー各社で主力となる働き盛りの年齢層だ。言い換えれば、かつてアスキーをはじめとする各社でMSXを推進していたスタッフたちと同じ年齢になったということだ。

```
MSX BASIC version 3.0
Copyright 2002 by MSX Association
Copyright 1988 by Microsoft
23435 Bytes free
Disk BASIC version 1.0
Ok
```

MSX-BASICの起動画面。クレジットに注目。

西が書いたMSXPLAYerのロゴ。

複雑な権利関係をクリアする

MSXPLAYerを公のものにするために、最も重要なのが権利関係の処理だ。どんなに完成度が高くとも権利関係がクリアされていなければ、一般公開は望めない。

今までMSXの権利は、商標はアスキーが保持していた。そもそも公式エミュレータの開発がスタートした時には、西 和彦はアスキーに取締役として在籍していたこともあり、名目も「アスキー公式エミュレータ」だった。しかしその後、西はアスキーを退社し、MSX関連の権利は西が引き継ぎ、あらためてMSXを推進していくこととなる。

もう1つ、MSX-BASIC、BIOS、MSX-DOSなどのソフトウェアはマイクロソフトとアスキーの共同の著作物である。事前交渉は数回行われたようだが、最終的には西が2002年の「東京ゲームショウ」でXboxの発表のために来日していたビル・ゲイツと会い、MSXの再活性化の方針の合意をもらい、移行契約を取り交わすことになった。これで、晴れてMSXPLAYerを表舞台で発表することができるようになったのだ。

メモ描きからMSXPLAYerのロゴが誕生した

MSXPLAYerの正式なロゴは、本誌が初公開となる。MSXシリーズには、MSX、MSX2、MSX2+、MSX turboR、その他多くのロゴがあった(※6)。MSXPLAYerにおいてもロゴが重要であることは言うまでもないが、これは西 和彦がホワイトボードにサラサラと描いたものが元となっている。

デジカメで撮影されたラフの画像を渡されたMSXユーザーグループの面々は、最初なんとも妙なデザインに少々困惑したという。特に、左右から飛び出た線をどうデザインすればよいかの議論が交わされた。そして奮闘努力の末、本誌での正式版の公開となった。

※6　MSXのロゴは、すべて西 和彦の発案によって作られてきた。ちなみに、最初のMSXのロゴは、VHSのロゴからインスピレーションを得たという。VHSのようにMSXが家庭に1台以上置かれるようにという意味合いがあったようだ。

MSXマガジン版MSXPLAYerの仕様が決められた

MSXPLAYerは、本誌に先行して「プロジェクトEGG」などにも提供されていた(※7)。MSXマガジンで初のお披露目となったのは、正しくは「BASIC対応版MSXPLAYer」だ。MSXPLAYerの特徴として、エミュレータに不慣れな初心者にも親しみやすくするために、スキン機能を実装している点がある。最初のMSXPLAYerは、グレーを基調とした地味なものであり、またボタンは、テストを兼ねてさまざまなものが検討されていた。

本誌収録バージョンでは、これらをもとに読者の使い勝手を考えて新たにスキンを作成している。3人のデザイナーが1つずつスキンを作り、MSXアソシエーションからも1点スキンが用意された。こうして、本誌には4種類のスキンが掲載されたわけである。

ボタンについては、機能の必要性を判断し追加や削除を行った。たとえば、サウンドの

※7　プロジェクトEGGに提供しているMSXPLAYerでは、すべてのボタンが外され、代わりにプルダウンメニューによる機能選択が可能になっている。

開発中のMSXPLAYer。スキンの色はグレーで、いかにも開発バージョンというもの。

音量スイッチは「ON/OFF」しかなかったが、「1/2/3/OFF」の調節を行えるように変更した。このあたりは、今後ユーザーの反応によってまだまだ改良されるべきところだろう。

MSXアソシエーションによると、将来スキンは、ユーザーが自由に変えることができるようになるという。現状でも画面やボタンのサイズ、ボタンの表示／非表示は、簡単に変更が可能である。そのあたりの柔軟性は最初から考えられていたということだろう。

MSXPLAYerの今後のロードマップ

MSXは、メーカー撤退以降、ユーザー自身によってその方向性が決定されてきた。MSXPLAYerは今後どこに進んでいくのか。誰かが決めるのではなく、これまでどおりユーザー自身が考え世に問うていくことになるのだ。MSXアソシエーションはユーザーの意向を反映させるべく、ユーザーとメーカーを橋渡しする組織だ。従って現時点でMSXアソシエーションに今後を尋ねてみても、「まずは読者からの反響を見てみないことには、次の段階を言うことはできない」との答えが当然のように返ってきた。

なお本誌は、現在のバージョンをまだ完璧に近いものとは見ていない。使い勝手にしてもまだまだ改良の余地がある。さらに、マルチプラットホーム対応の戦略上、より多くのコンピュータで動くようにする方が先決といった意見もある。編集部では、多くの取材から、ここしばらくは右の項目の実現に向けて動いていると予想している。

MSXPLAYerロードマップ

1 MSX turboR規格の実装

現在MSX2+相当の機能を実現しているMSXPLAYerだが、MSX turboR規格にも対応させるというもの。機能的には、R800 CPUのエミュレーションとMSX-DOS2の搭載、8bit PCM音源を追加する。これらはすでに搭載可能とのことだが、速度の点で課題が残っているようだ。基本的には、公開目標時点で販売されているWindows機のエントリーモデルにおいて速度的に正常動作させるべく開発が進められている。

2 ユーザーが自由にスキンを変えられる機能の実装

ユーザーオリジナルのスキンやスキンなしの状態も含めて、いろいろなスキンにワンタッチで切り替えることのできる機能。MSXを実行する画面サイズの自由な変更やボタンの配置なども任意に変更できるようになる。さまざまなデザインや画面サイズのスキンが登場すれば、それだけMSXPLAYerは楽しいものになるはずだ。

3 自作ソフトをカプセル化する方法の公開

自作したプログラムを配布するときに、BASIC版のMSXPLAYerを起動してロード、実行させるのではなく、EGGの市販ゲーム版のように、ワンパッケージにして配布できるようにするというもの。これは自作ソフトを手軽に他の人に配布して遊んでもらうための機能だ。

4 MSXPLAYerの機能拡張が行えるプラグイン機能の実装

MSXPLAYerの機能を簡単に拡張できるようにする機能。たとえば、メモリを512KBや4MBに差し替える、他の拡張音源を使えるようにするといったモジュールが考えられる。そのほか柔軟にMSXの機能を拡張できれば、さまざまなニーズに応えることができるようになるだろう。そのためにもMSXの基本機能の正確な再現が必須となる。

5 開発者向けMSXPLAYerの公開

ICE機能を実装したMSXPLAYer。加えて、他の環境のチェックがWindowsでできるバージョン。たとえば一部のPDAは斜めキーが効かないなどの制限がある。従って、今後MSXPLAYer上で動作するプログラムを開発するときには、「異なる環境でも問題なく動作するプログラム」として作っておく必要がある。こうしたプログラムを作りやすくするために、Windows以外の動作環境がチェックできる仕様を実現しなければならない。

MSXの明日

さて、まだまだやるべきことは山積みだが、MSXPLAYerのこれからを考えるには、MSX全体の未来を考える必要があるだろう。今後のユーザーの動向については、MSXマガジンも見守っていくつもりだ。MSXPLAYerの登場によって、参入するメーカーもあるかもしれない。しかし、これまで同様に明日のMSXを切り開くのは、ユーザー自身の創造的なイマジネーションである。ぜひ読者の皆さんも、MSXPLAYerの動作レポートや感想をMSXアソシエーションに電子メールで送っていただきたい。MSXPALYerの完成度を高めるのは、ほかでもないキミなのだから。

MSXアソシエーションからのメッセージ

みなさま、お待たせいたしました。やっとWindows版のMSXPLAYerを一般公開できました。
次の目標は、MAC OS、Linuxなどへの対応です。その次は、USB対応ROMカセットリーダーの作成、そして「次世代ワンチップMSX」の実現です。このチップをメーカーに採用してもらい実機の生産を再開することは、私たちMSXユーザーの夢です。これまで以上に困難であると予想できますが、実現に向けて、お金を使わずに頭を使って、みんなでがんばっていきたいと考えています。

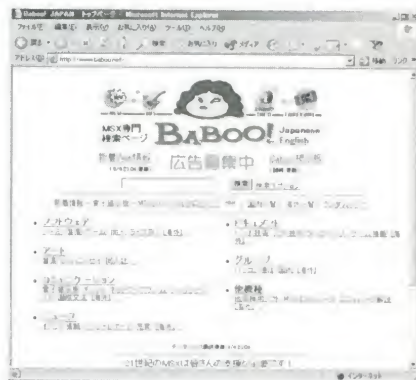
訪れておきたいMSXサイトを紹介しちゃう！

世間的には過去のモノとなってしまったかのようなMSXだが、インターネットの普及にともなって、MSXサークルやMSXファンの活動が逆に目立つようになってきた。これらに訪れるたび、嬉しくて涙がちょちょ切れるぜ。サイトの種類としては、ニュースサイト、データ検索サイト、ハード開発サイト、ゲームソフト開発サイトなどいろいろだが、どこもかしこも熱い、熱すぎる。ここでは「MSXサイトと言えば！」に続く、そんな押さえておきたい定番サイトを紹介しておく。国内サイトはもとより海外サイトにも大注目だ！

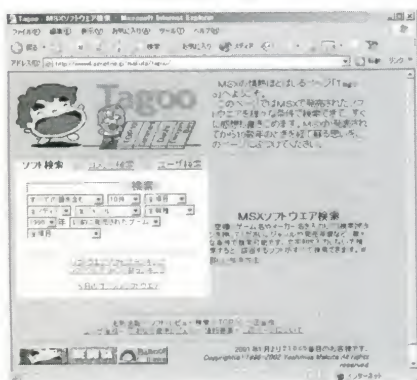
Illustration：常田朝子

JAPANESE SITE ▶▶▶

MSX関係のニュースを確認
Baboo!JAPAN

http://www.baboo.net/

あらゆるMSXサイトはここに集う、というイキオイで運営されているのが「Baboo! JAPAN」だ。1995年の2月から開設されている長寿サイトだけあって、登録サイト数は他の追随を許さない。現在も、月に数件のサイトが新規登録されている。もし新たにMSX系サイトを立ち上げたのであれば、ここへは是非とも登録しておきたいところだ。本サイトはリンクばかりでなく掲示板も盛況である。MSXの故障などについても、パワーユーザーが親切に対策を教えてくれるので積極的に利用したい。ところで本サイトの名称だが、Yahoo! JAPANをモジっていることは言わずもがなだが、いったいBabooとは何なのか？　実はコレ、MSX専門誌「MSX・FAN」(1995年に休刊) の名物パーソナリティに「バボ」女史という方がいて、その名称に由来しているのだ (タイトルロゴのふくよかな顔がそれ)。したがって読みも「バブー」ではなく、正式には「ばぼ〜」と読まなければならない。このことはサイトにもちゃんと記載されているので、多少強引でも「ばぼ〜」と読んであげましょう。なお、次に紹介する「Tagoo」は、本サイトの創設者である幕田氏が後に立ち上げたサイトである。

市販ゲームならおまかせ
Tagoo

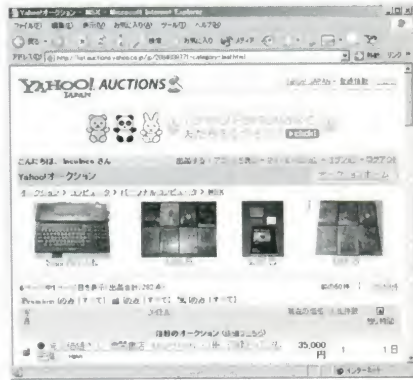
http://www4.airnet.ne.jp/makuta/tagoo/

MSX用に発売された市販ゲームを検索したり、ユーザーが自由にコメントを書けるサイト。タイトル下の鍵盤状のメニューを押すことで、「MSXソフトウエア検索」「Mマガ誌上TOP10情報」「登録コメント検索」「ユーザ検索」「伝言板」といったページに切り替えることができる。初めての人は、まずトップページ（MSXソフトウエア検索）で検索語を入力せずに［検索］ボタンをクリックしてみよう。多少時間がかかるが、これで市販ゲームが50音順で表示される。次にゲーム名をクリックすれば、より詳しい情報やユーザーのコメント一覧が表示される。ちなみに2002年8月現在のコメント数ランキング1位は「グラディウス2」で、アクセス数ランキング1位は「ソリッドスネークメタルギア2（コナミ）」となっている。マイナーなゲームの中にもコメントがたくさん寄せられている場合もあるので、こうした隠れ人気ゲームも探してみてほしい。サイト名の「Tagoo」だが、これはMSXマガジン初代編集長の田口氏の名前を拝借したものだ。また、タイトルロゴのキャラは、漫画家の桜玉吉氏デザインによる田口氏をモチーフとしたMSXマガジンのマスコットキャラ「こたぐち」である。

国内最大のMSXイベント
MSX電遊ランド公式サイト

http://www.msx-denyu.com/

東京・秋葉原で開催されたイベント「MSX電遊ランド」の公式サイト。1999年から年1回ペースで開催されている。そもそもは1996年から開催されているモオソフト主催の「MSXフェスタ」が発端だ。最初のうちは、一般の会議室やホールを借りて開催しており、規模も小さいものだった。しかし、2001年（MSX電遊2001）からは、ついに秋葉原の表通りに面した1階での開催となった。その立地を生かして、一般人をも誘い込もうと、入り口付近にゴザを敷き、ちゃぶ台、テレビ、ザブトンそしてMSXをドーンと置いて、アットホームさを演出。このユニークな企画は、来場者にはなかなか好評だったようである。本イベントではMSXゆかりの著名人による基調講演が恒例となっており、本書収録のMSX PLAYerもまた2000年の基調講演にて西和彦氏が発表。コンピュータ界に少なからずの衝撃を与えることとなった。また、こうした甲斐あって、電遊ランド2001では2日間の開催で、ついに入場者数6500人を数えるまでの盛況ぶり。本サイトには、電遊ランドに関するプレスリリースや新着情報が掲載されているので、今後イベントに参加したい人は確認してみるといいだろう。

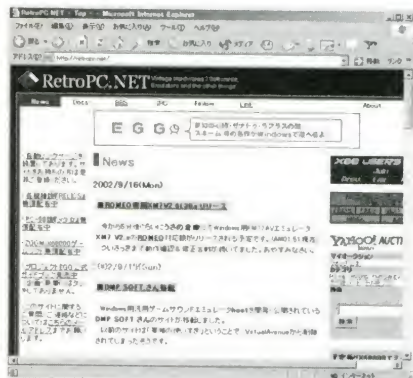
掘り出し物をチェック
Yahoo!オークション・MSXカテゴリ

http://list.auctions.yahoo.co.jp/jp/2084039771-category-leaf.html

オークションサイトとしては日本最大手の「Yahoo!オークション」。8ビットPCとしては、唯一MSXだけがカテゴライズされており、常に300点以上のMSX関連品（本体、周辺機器、ソフトなど）が出品されている。オークションであるから、本来は入札への参加が基本だが、ボ〜ンヤリと眺めているだけでもかなり楽しめる。ときにはびっくりするような珍品が出たりするから、MSXフリークとしては貴重な情報源として、ときどきはチェックしておきたい。出品内容だが、その多くはゲームソフトだ。かつては4〜5千円していたものが、一部のプレミアム品を除けば、せいぜい500〜2000円程度で落札できる。一方本体も少ないとはいえ、必ず出品されている。MSXの最終形態であるMSX turboR（A1ST）あたりは、相場としてもやや高目だが、それでも1万円ぐらいから落札できていることもある。MSX2程度のマシンならば落札価格は数千円（しかも前半）だから、本物のMSXを是非使ってみたい！ 昔欲しかったけど買えなかった〜！ なんて人はどうぞ。逆に、押入れにMSXを眠らせている人は、世のため人のため、MSX大好きな人のために、オークションに放出してくれると嬉しい限りだ。

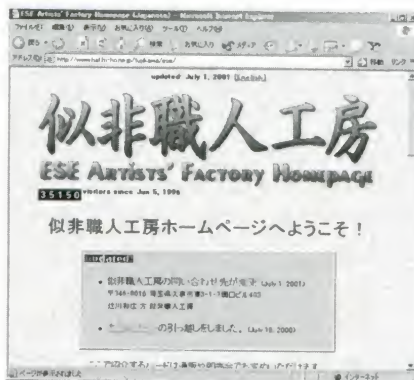
MSX Web探検隊

JAPANESE SITE ▶▶▶

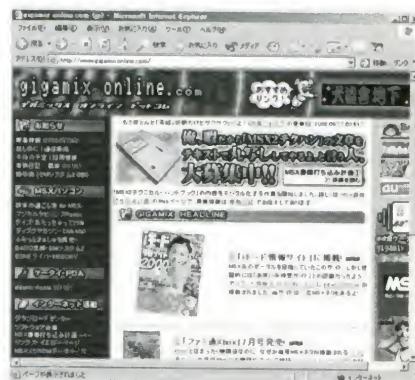
レトロPC情報ならココ
RetroPC.NET

http://retropc.net/

MSXの他、PC-8801/9801、X68000、FMシリーズといったレトロPCに関する情報を掲載しているサイト。トップページには、エミュレータ関連のニュースや、レトロPCゲームなどの復刻に関する情報が、ほぼ毎日アップされている。エミュレータというとアングラなイメージを持つ人もいるだろう。しかし、このサイトを運営しているRetro PC Federationは、ゲームメーカーや、ゲームサークルを相手に、過去に製作されたソフトウェアの配布を支援しており、決して裏サイトではない（具体的には、エミュレータの開発支援や、関連情報を扱うサイトへの領域提供、配布ソフトウェアに添付するマニュアルなどの製作代行がメイン）。これまでの成果としては、ボーステックの「RELICS」各機種版や、ZOOMのX68000版ゲームソフトの無償公開などが実現されている。また「究極!!X68000エミュレータ」（秀和システム）の執筆も手がけている。本サイトは1999年12月に登場し、2000年4月1日および2002年にリニューアルしている。工事中のコンテンツがあるものの更新は頻繁にされているので、最新の情報を得たいのであればブックマークだ。ともかくレトロPC好きには大変重宝するサイト。

SCSIカートリッジが買える
似非職人工房

http://www.hat.hi-ho.ne.jp/tujikawa/ese/

MSXでSCSI機器を使用できるようにするインターフェイス「MEGA-SCSI」や、MSX規格目一杯までメモリが実装できる4Mバイト増設RAMカートリッジ「ナイスメモリ!うっかりくん」など、目がうるうるモノのハードウェアの設計・製作・販売を行っているサイト。かつては実現できなかった拡張機器の開発を同人活動でやってるのだから恐れ入る。たとえば「MEGA-SCSI」があれば、合計7台までのハードディスク、光磁気ディスク（MO）、Zipドライブ、CD-ROMが接続でき、さらにエプソン製イメージスキャナなども使えるそうだ。本サイトでは、こうした製品の通信販売も行っている。プロジェクトは他にもあって、特に現在進行中の「似非PLD」プロジェクトは興味深い。これはMSXを、プログラム可能な論理素子「PLD」デバイスを使って再現しようというもの。言わば、MSXをゼロから作り起こそうという計画だ。PCによるエミュレータと違って、単独の基板だけでMSX相当の動作が可能となる。すでにMSX1相当の再現は完了しており、今後の予定としてMSX2相当の再現が予定されている。最近は、PLDデバイスそのものが進化しているとはいえ、すごすぎ～る！

ゲームもスゴい元気サークル
gigamix online

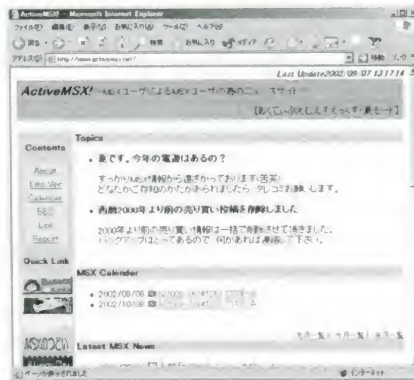
http://www.gigamixonline.com/

MSXサークル「野望集団GIGAMIX」のサイト。「マジカルラビリンスRemix」「クイズあたっちゃって25%」など、メチャクチャ出来がグレートなゲームを発表している。しかも、これらのゲームで使われている“マルチメディアもどきソフトウェア簡単制作開発支援ツール「DMシステム2」”も公開中。このシステムは「終末の過ごし方（アボガドパワーズ）MSX移植版」などのゲームでも用いられているなど、実績も十分である。DMシステム2についてはテクニカルハンドブックも通販されているので、興味を持ったなら入手しておきたい。本サイトでもう1つ紹介しておきたいのが「gigamix mobile」というケータイ向けコーナー。ここには「MSX2起動画面」や「intel outside」などの、MSXユーザーならニンマリしてしまう待ち受け画面のダウンロードサービスがある。また、MSXのハード／ソフトを取り扱っている中古ショップや、FDDベルトの入手方法といった、お得なミニ情報も掲載されている。いつでもどこでも、MSXにドップリ浸り、心のうちに眠るMSX魂を燃焼させたい人にうってつけだ。ついでにゲストブックと事務日誌（日記）も好評である。マメにチェックしてMSXのツウになれ！

オカルト系カルトメーカー
ソフトスタジオWINGファンクラブ

http://sakai.cool.ne.jp/souzuka/

「ソフトスタジオWING」公認のファンサイト。MSX草創期に「白と黒の伝説」というオカルト系AVGを発売し、その後も、霊、超能力、異星人、古代文明といったテーマのゲームソフトを発売し続けていたメーカーだ。WING自体は、活動の再開/休止を何度か繰り返しているのだが、少なくとも存続はしている(しかしながら現在は休止中)。というわけで、本サイトの中心はWINGに関した情報である。当然、WINGが発売した全ソフトの一覧もあるし、スタッフの紹介もある。WINGの歴史もちゃんとおさえてある。主要スタッフの一人である宗野氏へのインタビューも興味深い内容だ。BBSへの参加も自由。WING好きは行ってみるベシ。そして、WINGのゲームソフトに興味がわいたのであれば、EGG(p.70参照)に行ってみるとよい。そこで「白と黒の伝説」(MSX版)が購入できる。ちなみにこのゲームだが、音楽機能の貧弱だったMSXで豪華なBGMを鳴らすため、別途音楽テープが付属していた。しかも、画面の指示にしたがって、ユーザーが音楽テープの再生・停止操作をするようになっていたのだ。なんとも力技、それでもゲームは2万本をセールスしたというのだからオドロキだ。

ユーザー参加型情報サイト
ActiveMSX

http://www.activemsx.net/

BabooJAPANがMSX最大のポータルサイトなら、こちらはMSX最大のニュースサイトだ(もっともMSX関係のニュースは、今となってはそう多くないので、更新頻度はボチボチである)。便利なのは、過去記事が完全保存されているところ。もちろん参照も簡単だ。ニュース右下の、イベント、ソフトウェア、ハードウェア、コマーシャル、レポート、募集中といったカテゴリ別のリンクをクリックすれば、過去記事が表示される。たとえば「ハードウェア」や「レポート」を見てみれば、実機を持っているユーザーがたいてい直面する、MSXのFDD用ドライブベルトに関する情報などもすぐ発見できる。どこかのBBSで尋ねる前に必ずチェックしておきたい場所といえる。コーナーとしては他にも、MSXに関する「コラム」や、MSXのソフト・ハードに関する「売り買い情報」などがあるのだが、ニュースも含めて本サイトの特徴は、ユーザーが参加できる点だ。各コーナーのための新規投稿リンクをクリックすればフォームが現れるから、誰でもすぐ投稿できる。スケジュールの参照・投稿ができるカレンダーや、BBSも設置されているので、イベント告知や仲間集めなどにも利用したい。

牛の歩みでモ~オソフト
モオソフト

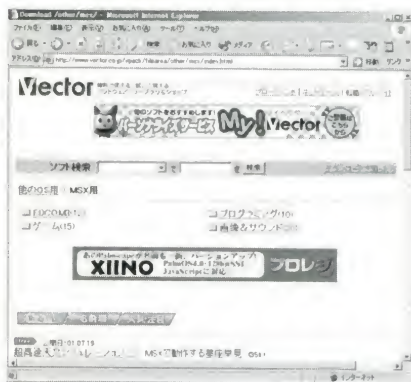
http://www.din.or.jp/~mosoft/

MSXサークルというより、MSX活動を支援する団体として始まったモオソフトのサイト。ファミリーソフトという発売元でMSXのオリジナルタイトルを発表していた漫画家の青井大地(青井泰研)氏や、漫画家、プログラマが中心となって設立された。ここでは現在、初心者向けのMSX解説書籍「MSXアルハンブラ」やソフトウェアの販売、「MSXフェスタ」などのイベントの開催といった活動をしている。サイトで購入できるMSX版ソフトウェアの中では「MSXトレイン2号」というディスクマガジンと「SMレディ」がオススメ。なかでも「トレイン2号」は、発売当時にアクシデントがあり入手しにくかったといういわくつきの品だ。そこに含まれる「神々の宇宙船」というゲームは完成度がとても高く、一度はプレイしないとソンかも。一方の「SMレディ」は一発芸的な同人ソフトなのだが、漫画家の渡辺健一氏による豪快な作品で、MSX版同人ソフトが満喫できる。モオソフトは「原則として品切れを起こさない」というポリシーで活動しているため、MSX系の即売会では、いつも膨大な数の同人誌等を陳列しているのが特徴。机に山のように本が積まれていたら、それはたぶんモオソフトだ!

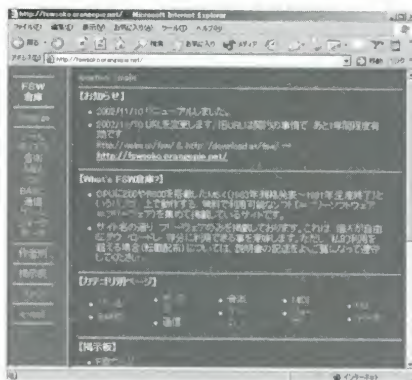
MSX Web探検隊

JAPANESE SITE ▶▶▶

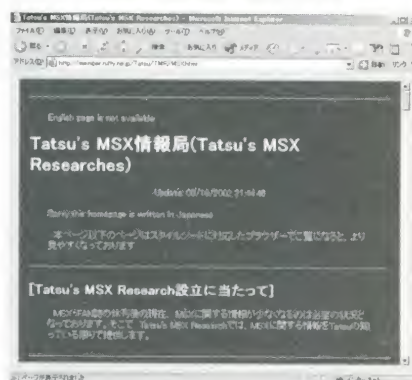
手軽にライブラリが落とせる
Vector (他のOS用>MSX用)

http://www.vector.co.jp/vpack/filearea/other/msx/

プログラムライブラリの老舗と言えばVectorだ。かつては、フリーウェア&シェアウェア集としてCD-ROM書籍「PACK」シリーズを発売していたが、現在はインターネットをメインに活動中だ(1999年に発売された「PACK for WIN」は、なんとCD-ROM12枚組みという途方もない物量だった。あれはタマげた)。Vectorに登録されているのは、主にWindows、Mac、PDA用ソフトである。しかし、目をよ〜く凝らして見るとMSXのカテゴリも健在。MS-DOS、OS/2、FM-TOWNS、X68000よりも、さらに深い階層にあるため見つけにくいが、それでも現在70本ほどのMSX用ソフトが登録されている(基本的にフリーソフト)。種類としては、プログラミング関係、画像&サウンド関係、そしてゲーム関係など。新しめのソフトはMSX-DOS環境用のものも多い。まとまったソフトとしては拡張BASIC「EDCOM3」(杉山夏樹氏作)がある。これは縦書きテキスト表示など独特の機能を搭載し、ディスクマガジン「FFP」などでも使われていたという実績がある。以上、Vectorは利用しやすいサイトなので、たまに眺めておくと運命的な出会いがあるかも。願わくば、もっと登録が増えてほしいものだ。

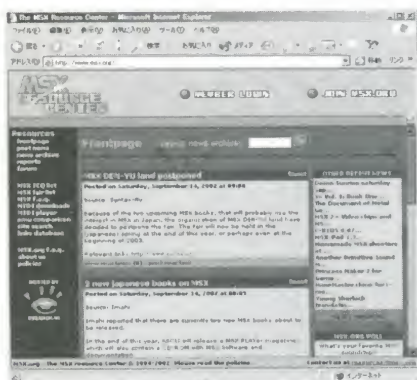
選り優りのソフトを保管
FSW倉庫

http://www.interq.or.jp/white/white/fsw99/findex.html

MSXというとゲームマシンだと捕らえている人もいるが、MSXにはクリエイティブ魂を刺激してくれる魅力もある。現にMSXでゲームプログラミングや、お絵描き、作曲に目覚めた人も多い。こうした用途で重要なのがツールだ。本サイトは、主に各地のBBSにあったツール(フリーソフト)を作者の許可を得て公開しているサイトである。経緯的には、インターネットの普及により各地のBBS(パソコン通信のホスト)の閉局が相次ぎ、退避的に登録されるソフトが多くなった。登録数は140本ほどで(2002年10月現在)著名なツール類はだいたいそろっている。MSXでそれなりのツールが欲しければ、まずここでソフトをダウンロードしてみよう。ただし、古いソフトもあるので、作者に連絡がつかなかったり不明な点があっても、付属のドキュメントで自己解決するしかない可能性が高い。そこだけは了解しておこう。それとFWS倉庫はダウンロードは自由だが、新規登録は管理者(たろさん氏)が手動で行っているため、メールで依頼すること。なお、FWS倉庫で特にお勧めなツールについては「定番オンラインソフト」(p.140)でも紹介しているので参考にしてほしい。

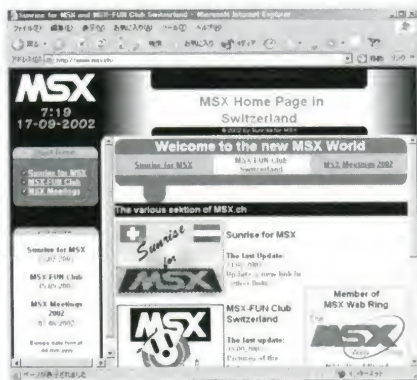
コアなMSX野郎の開発指南
Tatsu's MSX情報局

http://member.nifty.ne.jp/Tatsu/TMR/MSX.html

東北のパワーユーザーTatsu氏のホームページ(http://member.nifty.ne.jp/Tatsu/)の1コーナー。MSXの情報ページとしては珍しく、C言語に関する詳しい情報が多数載っている。MSXとしては代表的なMSX-Cの他、無料のSOLiD CおよびHI-TECH Cの情報も豊富(MSX-Cは、作成したプログラムの配布/販売をするのにライセンス取得が必要だったという側面から、他のC言語への関心も高かった)。各コンパイラの動作状況やバグ情報、自作のライブラリなどの突っ込んだ内容が多く、実践的かつマニアックと言える。どこを見ても字ばかりのページではあるが、MSXで本格的な開発をしたい人にとっては、きっとさまざまな発見があるはずだ。一方、初心者ならばトップページの下にある「リストのページ」を見てみよう。ここにはBASIC言語1行でできた「重力逆転!!」および「N-DRIVE Part2」というゲームが掲載されている。たった1行の妙技には感心するばかり。入力時にコツが必要な部分もあるが、ぜひ自分で入力して楽しもう。それと、MSX-DOSで動く優秀なエディタ「テキストエディタV&Z」(宇佐美仁識氏作)が入手できるのは本サイトだけなので、これもダウンロードだ。

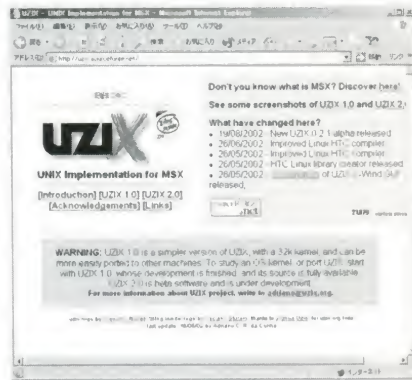
海外のMSX系ニュースサイト
MSX Resource Center

http://www.msx.org/

URLを見ると、あたかもMSXの公式サイトのようであるが、実はオランダのユーザーサイト。日本国内のどのMSXサイトよりも頻繁に更新されていて、あなどれない。あくまでも英語であるが、ニュース投稿や過去記事を閲覧することもできるし、MSXに関するFAQも大変充実している。MSXなマニアならいっぺんはのぞいておいて損はないだろう。驚くべきことは、なんと日本国内のニュースも詳細に報告されているところ。例えば「コナミがJ-PHONEでゲームを配信」なんて記事があったりする。これは、日本に留学している仲間からの情報なのだそうだ。MSXは、全世界で400万台以上出荷されているが、そのうちの半分が日本で、残り半分は海外である。したがって、MSXのファンは海外にも非常に多く、その情熱もかなりのモンなのである。ちなみに国別で言うと、MSXユーザーが活発なのは、日本、オランダ、そしてブラジルなど。そのため、ニュースの中には「ポルトガル語版スナ○チャー開発中」なんていう、ちょっぴり危険な香りのする記事もあったりしてわくわくだ。英語は苦手、という人でも文章は比較的簡単なので、気合いを入れて回ってみてはいかが？

海外でMSX拡張ハードウェア
Sunrise swiss

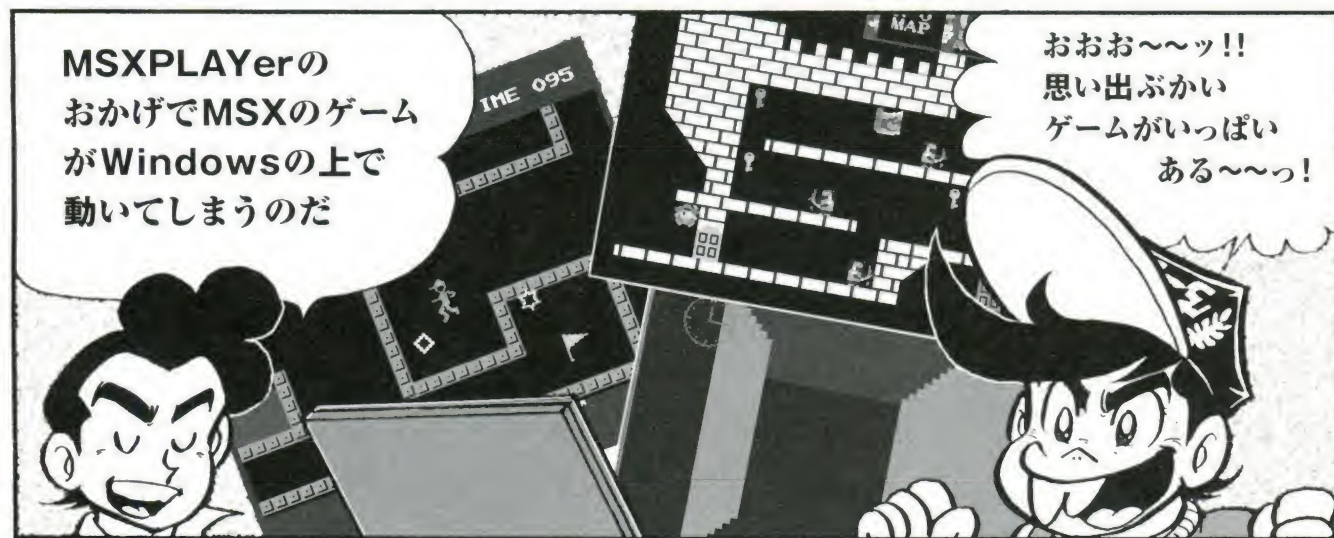
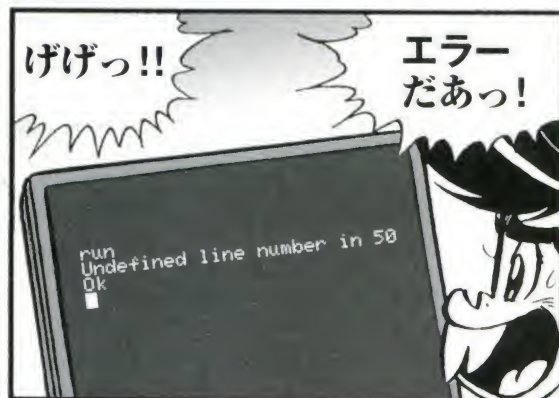
http://www.msx.ch/

日本の「似非職人工房」のように、同人規模でMSXに関係するさまざまなハードウェアの開発と販売を行っているサイト。具体的な製品としては、IDEハードディスクを接続するためのカートリッジ、RS-232Cカートリッジ、IDEとRC-232Cのちゃんぽんカートリッジ、ヤマハのOPL4サウンドチップを搭載したFM音源＋PCM音源が使えるカートリッジなどがある。海外では、いわゆるPCケースにMSXのマザーボードを入れて、今風のマシンに改造するといった試みもよく行われているのだが、本サイトにはそのための「MSXのスロットを増設するハード」も売っている（かつてはPS/2キーボードをスロットに繋ぐアダプタなども存在した）。さらに、かつては次期MSXに搭載予定だったはずの幻のVDP「V9990」増設カートリッジなんてものまで販売していたこともあった。ともかく海外ユーザーのパワーは我々の想像以上だ。ちなみに、この中でもっとも興味がわく製品はIDEカートリッジだと思うが、これについては日本語での解説がある。また、値段は47.65ユーロ（日本円にして5500円ほど）だ。注文は、日本語ページがあるのでHDDを増設したいなんて人は行ってみて。

MSX版のUNIXが誕生！
UZIX公式サイト

http://uzix.sourceforge.net/

MSXでUNIX。これだけ聞くと「なんだそれぇ〜」って感じである。しかし、海外では「UZIX（うじっくす）」という、MSXでUNIXライクなOSを動かそうというプロジェクトが、かなりマジメに進められているのだ。UNIXといえばネットワークであり、ネットワークといえばインターネット。UZIXでは"ダイヤルアップ＋シリアル接続のみ"という苦しい構成ではあるものの、しっかりインターネット接続まで実現できているという。しかもFudeBrowZerなるUZIX用Webブラウザまで存在しているというのだから夢が膨らんでしまいそう（ちなみに日本語表示にも対応済み）。とはいえ、今のところ実用性は今一歩。MSXのパフォーマンスが低いこともあるし、FudeBrowZerの画面もケータイ電話並みである。したがって、あまり過度な期待はしないほうがいいと思うが、将来的にはちょっと楽しみである。なお、本サイトは英語オンリーだが、邦訳サイト*もあるので、詳しい内容はそちらで確認してね。ところで、肝心のMSXPLAYerでのUZIXの動作についてだが、UZIXそのものは動作する。しかし、MSXPLAYerはシリアル接続をサポートしていないため、残念ながらインターネットへの接続はできない。

＊ UZIX邦訳サイト：http://www.nodus.ne.jp/~ghost/msx/uzix-j.html

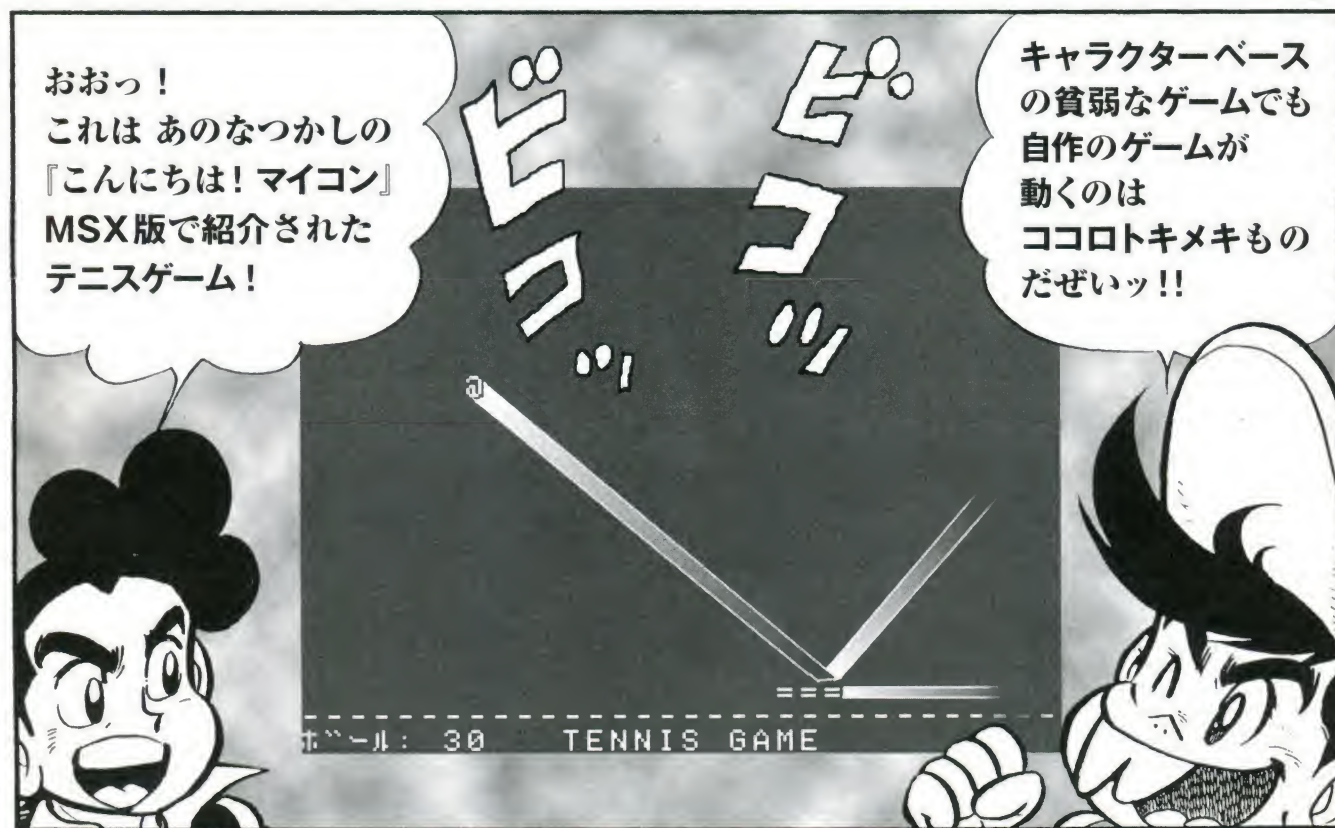
↑作者所有のASCII STICK II Turbo

「マジ〜？ あらしがMSXPLAYerを使ってるなら、ボクも〜」

すがや先生お手製のテニスゲームは、本誌付属のBASIC版MSXPLAYerに「TENNIS.BAS」として収録しているので遊んでくれぇ！タマには!!

おおっ!
これは あのなつかしの
『こんにちは!マイコン』
MSX版で紹介された
テニスゲーム!

ピコッ
ピコッ

キャラクターベース
の貧弱なゲームでも
自作のゲームが
動くのは
ココロトキメキもの
だぜいッ!!

ボール: 30 TENNIS GAME

おわし

【MSXPLAYerあらしが読めるのはMマガだけ〜。おそらく】

Mマガおたよりコーナー

お互い年くったなぁの巻

前号から10年もたっているらしく、昔のおたよりもヤギが食っちゃったし、しょうがないので、今流行のインターなんとかのうぇっぶなんたらで（どうせMSXはつながらんよ。しくしく）ちょっくら募集してみたのだ。そしたら来ました、すぐ集まりました。Mマガの小さなかけ声にジャカスカドンドカおたよりのオンパレード！ うおお、ありがてぇよ。持つべきものは読者なりの感激うるうるおめめキラキラ、さあ、ひさかたぶりのおたよりコーナーだ。ハリキっていってみよーっ！

Mマガの復活ですか？ メール等で知りましたが猛烈に感動しております。そもそもパソコンというものに触れる第一の機会はMSXでしたからね。ゲームはもちろんBASICを使い自分でゲーム等を作ることができてとてもおもしろかったですね。今、家にはFS-A1ATが眠ってますが、おそらく故障して動かないでしょう。そこでMSXPLAYerには大変期待してます。懐かしのゲームを再び見ることのできるときを。

本靖由さん（京都府）

MSXマガジンは一度も死んでないので本当は復活ではないのだ！ 1992年夏号の次に出る予定だった秋号の発行予定が大幅にずれただけなのだ！ というわけでお待たせしましたMSXマガジンいかがでしたでしょうか。

（負けず嫌いな編集者）

はじめまして、happy new MSX year!!

mochida,akira（長野県伊那市）

こちらこしょ、はじめらして。旧年中はいろいろとおしぇわになりましゅた。今後どみょMSXをよろしくお願い申し上げまーしゅ。う〜いっ。

（早くもお正月気分の編集者）

初めまして。MSXマガジンが復刊されるのですね。私がパソコン系に触れたのは、MSXが初めでした。MSXマガジンも当時よく購読してましたが、MSXのブーム（？）が終わりスーファミへ、それからニンテンドウ64、プレステ、そして今はパソコンにハマッてます。でも、まだ当時のMSXのハードと数本のソフトは処分できず持っています。ネットでMSXファンのいることがうれしかったですし、MSXマガジンの復刊。必ず、買います。アスキーさんガンバッテください。

ヒロカメラマン（大阪府）

復活おめでとうございます。創刊から買い続けていた者としては非常に感慨深いものがあります。MSXのせいで私は今プログラマという仕事に就いていると言っていいでしょう。ところで、XBoxはMSX3ですか？

ケロ（福岡県）

ありがとう。とりあえず廃刊宣言も休刊宣言も「絶対」しないので今後も出たら買い続けるように。次は半年後か1年後か、もしかしたらアスキー50周年記念企画あたりになっているかもしれないが。ところでXBoxについてだが、MSXマガジン2027（名称未定）あたりにXBoxPLAYerがMSXのひとつとして付録されていたらMSXの一員なのだろうし、入っていなかったら違うのだろう…きっと。というわけで、もうしばらく結論はお待ちください。

（きっとそのときも編集者）

ベーマガは今もって健在ですしP/ECEにワンダーウィッチと「コンピュータでモノ創り」の灯は今だ絶えていませんが、やっぱりいつまでも私にはMSXなのです（きっぱり）。というワケでMSXPLAYerを手に入れてモバイルPCでいつでもどこでもMSXライフするです。もう決めました…で、どうやったら買えるのかな？(.・) ← おぃ

かいるっ（愛知県）

この本を買っているなら、あなたはすでにMSXPLAYerを入手済みなハズ。さあ、素晴らしいMSXライフを。ところでモバイルPC向けにMSXPLAYerを考えるなら、タブレットPCの機能を活かすとか、Pocket PCの特徴を活かすとか、新しいチャレンジがいろいろ考えられますが、その後も携ゲホゲホッ、ポケットゲホゲホゲホッ、それからもうすぐゲホゲホゲホゲホゲホゲホゲホゲホ。はぁはぁ。とにかくいろんなMSXの楽しみ方ができるようになるような気がしないでもないのです。

（かなり風邪ぎみの編集者）

小学校3年生頃に最初のMSX（HB-10）を購入して、早15年以上経過していますが、今でもMSX（A1ST）は目の届くところに置いてあります。彼女（MSX）を見るたびにMSXマガジンを片手に戯れたよき時代を思い出します。思い出の中でしかなかったMSXマガジンが復刊するなんて世の中捨てたもんじゃありませんね。

STA（エスタ）（東京都）

昔から「すてる神ありゃ、拾う神あり」と申しまして。このたびは拾っていただきありがとうございます。願わくば今後ともMSXと戯れつづけてくださらんことを。

（捨て猫気分な編集者）

MSXマガジン、復刊おめでとうございます!! MSXとの出会いは小2のとき。あのときは、よもや自分の大学に入ったときの最初の友人グループが、MSXで盛り上がった仲間たちになるとは思いませんでした。そして大学生のときに、MSXの「MS」が "Micro

Soft"なのか、「松下・ソニー」なのかで後輩と軽い賭けをして、引き分けのままなのですが、Mマガ編集部的にはどちらの説でしょう？　それとも実はXBoxが真のMSXなんでしょうか？

so-so（千葉）

MSXについては「三菱(9)・サンヨー(15)・X」の略だという説と「ミツミ(20)・サムスン(26)・X」の略だと言う説もあったりして。なお、カッコ内はMSX規格におけるメーカーIDとなっております。ご参考まで。

（技術担当な編集者）

こんにちは、MSXマガジンの文字に思わず懐かしくなってしまいました。ふと思い出したのですがDante3とかどうなったんでしょう？

だんてんぐ（埼玉県）

時は足早に過ぎ去り時代は新しい世紀へと突入した。いつまでも昔のことにこだわらず、新しいMSXの未来を共に築こうではないか！　と、言って誤魔化したいところだけれど真面目な話、ツクール類については復活したいものだなぁ。応援よろしく！

（本当は投稿作品好きな編集者）

昔のMマガで「人工知能うんちく話」を連載されていた方は今どうなさっているのでしょうか？

Y.UME（ゆめドット）（東京都）

鹿野司先生はサイエンスライターとして活躍されております。今度ぜひまた本誌に書いてもらいたいですね。

（今も心は科学少年の編集者）

単なる同窓会でなく、前を向いて歩いていきたいですね。

定　翼（栃木県）

MSXPLAYerは、ただMSXを最新PC上に再現することを目指して作られたのではないそうです。でも何にしてもMSXを支えてきたのはユーザーのパワーですから、これから先を考え、切り開くのもユーザーの意思でしょう。たえず前を向いて行きましょう、お互いに。

（努力、友情、勝利な編集者）

復刊、おめでとうございます。気が早いようですが、来月号も楽しみにしております。あ、そういや、ポケットバンクの新刊とかそろそろ出ませんか？？？

Rabbit's（神奈川県）

来月って…。いや、月刊はちょっと。いろいろと事情がありまして。え？　ポケットバンク？　それもちょっと、えへへへ。

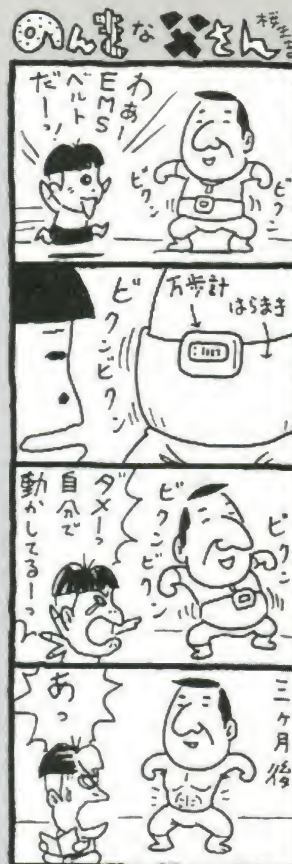
（大人の事情が渦巻く編集者）

Mマガといえば、BASICコンパイラ「ベーしっ君」。Mマガ休刊後くらいに電器屋さんで「ベーしっ君ください」と言ったら、「入荷次第連絡します」と言われたっきり音沙汰ないんです。そろそろ入荷しないんですかねぇ？　待ってるんですけど。

とっきー（大阪府）

電器屋さんへの入荷状況はわかりませんが、今回の売上げがよくて、次が出ることになって、リクエストが多ければ本誌に収録されるかもしれないかも。

（無責任で、すぽぽーんな編集者）

超軟門 MマガQUIZ!

自称Mマガ野郎を名乗るキミに、その実力のほどを確かめるべく、軟問クイズを2問出すぞ。全問正解なら文句なしにノーベル賞もびっくりのMSX大王だ。しかし1問正解だったときにはMSX平民、そして全問不正解ならMSX大貧民だ(精進が足り～ん)。てなわけで、**いざ真剣勝負！**

正解は次ページに☞

問題1 このMマガ歴代キャラの中で間違っているのはどれ？

Ⓐ MSX坊や

Ⓑ こたぐち

Ⓒ にせQ

問題2 これらのロゴの中でおそらく正しいものはどれ？

Ⓐ

Ⓑ

Ⓒ

Mマガおたよりコーナー

もっともっとMSXゆかりの有名人を読んでください。とりあえず小島監督をお願いします。
コナミファン（神奈川県）

何!? 有名人を誌面に!? ふーむ…。で、いったい何を読んでもらいたいのかな？ もしかしてアレの企画書とか？
（実は私も聞きたい編集者）

復刊おめでとうございます。間に同人の「復刊準備号」がありましたが、アスキーからMマガが出る日がくるとは正直思っていませんでした。MSXが世紀を渡って生き残ったのは、多くの思い出を持つものとして嬉しいです。しかし「'92夏」の次が「'02冬」とは…。次も10年後？ せめて「北の国から」と同じくらいのペースで出てくれてたらなぁ…。でも、あれも終わっちゃったし。まさか、また世紀を超えるなんてことは…。
白神裕久（愛知県）

父さん、ようやくMSXマガジンが出ました。でも、あれから10年も経ったわけで、いつの間にか世紀も変わってしまったわけで、みなさん応援よろしくお願いします。
（個人的には同人版もわりと好きな編集者）

遠い昔、まだ10歳だった幼い私をてっきりお父さんは捨てたんだと思ってました。でも本当は違ってたんですね。お父さんは本当は、本当は私のことをずっと愛してくださっていたんですね。見て、お父さん。私、二十歳（はたち）になりました。もう立派な大人よ。これも親切なみなさんに大切に育てられ、かげからお父さんに支えられ、そして世界中のお友だちに応援していただいたおかげです。ありがとう！ さようなら…ってどこ行くんだよ！ MSX!!
元勇者様（東京都）

光陰矢の如し…ともあれ、あなたは最低の父親でした。と、プリメのエンディングで言われたときは、ワザとやったとはいえ、ちょっとショックでした。それはともかく、お父さんって誰だよ。はっきり言いなさい、お父さんは許しませんよ。
（こっちだって元勇者様の編集者）

おお、Mマガ復活ですか。これはめでたい。となると強敵は「ゲーム脳」ですな。
アンチβ派（神奈川県）

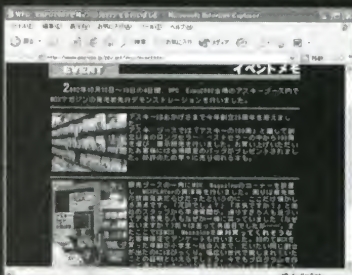
心配無用。MSXを極めれば、悟りの境地に至る也。
（元祖θ派な編集者）

MSXマガジンの復活が嬉しくてたまりません。気が早いようですが次号はいつ発売なのでしょうか。えっ、次号予告を見ろって、ああ本当だ。安心しました。
舛添耕一（北海道）

え、どこ？ どこ？ どこに次号予告ってあるの？ ああ本当だ。安心しました。
（ぜひ次号も手伝いたい編集者）

のんきな父さん 桜玉吉

光陰矢の如し
父さん
父さん
似るなっ!

●QUIZの答え 問題1→A（MSX坊やに髭はない） 問題2→C（AとBはおそらく違う）

ウーくんのソフト屋さん

●Illustration／桜沢エリカ　●Program／花岡朋和

かくれんぼゲーム

BASIC版 MSXPLAYerに収録
プログラム名：UOO-KUN.BAS

だれでも一度は遊んだことがある、かくれんぼ。鬼に見つかりそうになったときのドキドキ感や、鬼になってかくれている人を発見したときの達成感がたまりません。今回は、そんな"かくれんぼ"をゲームにしてみましたよ。しかも、なんと反射神経ゲームです。これはとっても忙しい！　ウキャーッ！

ゲームを実行する

このプログラムはBASIC版MSXPLAYerに収録されていますから、プログラムリストを打ち込む必要はありません。BASIC版MSXPLAYerを起動したら、プログラムを読み込んで実行しましょう。それには、MSXのBASIC環境で次のように入力してください。

```
RUN"UOO-KUN.BAS"[リターン]
```

ゲームで遊ぶ

画面に「Now Reading...」と表示されます。そのまま1分弱待つと、中央にミンミンちゃんが、周りに8本の木が現れます。「Push Space Key」と表示されたら［スペース］キーを押してください。いよいよゲームスタートです。

8本の木のどれかから、ウーくんがピョコリとまろび出てきます。反射的に、その木のある方向のカーソルキーを押してください。ミンミンちゃんがその方向に動きます。ウーくんが見えている間にキーが押されていれば得点です。ウーくんが木の陰に引っ込んでしまった後でキーを押しても得点されませんよ。

上下左右方向のウーくんを指すには、［↑］

[↓] [←] [→] のカーソルキーを1つ押してください。また、右上、右下、左上、左下方向のウーくんを指すには、カーソルキーを2つ押してください。たとえば、右下なら[→]キーと[↓]キーをほぼ同時に押しましょう。斜め方向を指すのは少し難しいですが、タイミングよく押してくださいね。

木から出てくるのはウーくんばかりではありません。ときどき女の子が出てくることがあります。この子はたまたま遊びにきていた近所の子ですから、この子に対してキーを押してしまうと減点になってしまいます。瞬時に、ウーくんか、そうでないかを判断して、ウーくんだったときにキーを押しましょう。

ゲームは約1分間続きます。その間画面の下には、見つけたウーくんの数と、間違えてしまった数が表示されます。そして、これらの数から得点が計算されます。果たして時間内にどれだけ得点できるでしょうか？　高得点目指してガンバって！

ゲームを終わりにする

ゲームを止めるときには、MSXPLAYerの[Reset]ボタンを押してリセットするか、MSXPLAYerを終了します。また[CTRL]+[STOP]キー(Windowsでは[Ctrl]+[PageUp]キー)を押すとBASICに戻ることができます。

ゲームを実行してしばらくすると、画面に「Push Space Key」と表示されます。[スペース]キーを押せば、早速ゲームが開始となりますけど準備はできました？

```
1000 '--- Initialize
1010 SCREEN5,2:COLOR15,12,12:CLS:DEFINTA-
Z:OPEN"GRP:"AS#1:A=RND(-TIME)
1020 PRESET(80,90):PRINT #1,"Now Reading..."
1030 FORI=0TO15:A$="":FORJ=1TO32
1040 L=PEEK(&HF6A4)*256+PEEK(&HF6A3)
1050 PRESET(105,100):PRINT #1,L
1060 READ B$:A$=A$+CHR$(VAL("&H"+B$))
1070 NEXT:SPRITE$(I)=A$:NEXT
1080 FORI=3TO15STEP2:COLORSPRITE(I)=1:A$="":FORJ=1TO16
1090 READ B$:A$=A$+CHR$(VAL("&H"+B$))
1100 NEXT:COLORSPRITE$(I+1)=A$:NEXT
1110 SPRITE$(18)=STRING$(32,255)
1120 SPRITE$(19)=STRING$(3,&HE0)
1130 FORI=0TO15:READ R,G,B:COLOR=(I,R,G,B):NEXT
1140 DIM X(8),Y(8),D(10),M(34):CLS
1150 FORI=1TO8:READ X,Y:X(I)=X:Y(I)=Y+2:NEXT
1160 FORI=0TO10:READ M(I):NEXT
1170 '--- start
1180 PUTSPRITE3,(120,86),,8
1190 PUTSPRITE4,(120,86),,9
1200 FORI=5TO16:PUTSPRITEI,(0,217):NEXT
1210 COLOR15,12,12:CLS
1220 FORI=1TO8
1230 LINE(X(I),Y(I))-STEP(15,35),11,BF
1240 DRAW"C10BM=X(I);,=Y(I);BU1BL8R31M-15,-23L1M-
15,23":PAINT(X(I),Y(I)-2),10,10
1250 NEXT
1260 COLOR8,12:PRESET(32,0):PRINT#1,"Uoo-Kun's Kakurenbo Game"
1270 COLOR15,12:PRESET(8,196):PRINT#1,"Hit: 0 Miss:
0":PRESET(8,204):PRINT#1,"Score:  0"
1280 DRAW"C15BM249,204D2L30U2D2L30U2D2L30U2D2L30U2"
1290 FORI=0TO9:D(I)=I*4+RND(1)*4:NEXT
1300 C=0:D=0:S=0:H=0:M=0:T=61:GOSUB1930
1310 CIRCLE(128,112),48,11,,,.35:PAINT(128,102),11,11
1320 COLOR 15,0:PRESET(72,100),,XOR:PRINT#1,"Push Space Key"
1330 IF STRIG(0)+STRIG(1)=0 THEN 1330
1340 COLOR 15,0:PRESET(72,100),,XOR:PRINT#1,"Push Space Key"
1350 CIRCLE(128,112),48,13,,,.35:PAINT(128,102),13,13
1360 PLAY"V12T16004L16CDE"
1370 PUTSPRITE3,(120,94),,8
1380 PUTSPRITE4,(120,94),,9
1390 ON INTERVAL=60 GOSUB 1930:INTERVALON
1400 '--- Main Routine
1410 N=0
1420 IF D(D)=C THEN D=D+1:F=1 ELSE F=0
1430 IF N<>0 THEN 1490
1440 P=RND(1)*8+1:LR=1+(RND(1)>.5)*2
1450 FORI=5TO12:PUTSPRITEI,(0,217):NEXT
1460 PUTSPRITE0,(X(P),Y(P)),11,18
1470 PUTSPRITE1,(X(P),Y(P)+16),11,18
1480 RR=RND(1)*4+1:RP=RND(1)*2+1
1490 X=M(N)*RR*LR
1500 PUTSPRITE5+F*4,(X(P)+X,Y(P)),,F*4
1510 PUTSPRITE6+F*4,(X(P)+X,Y(P)),,1+F*4
1520 PUTSPRITE7+F*4,(X(P)+X,Y(P)+16),,2+F*4
1530 PUTSPRITE8+F*4,(X(P)+X,Y(P)+16),,3+F*4
1540 N=N+RP:IF N>10 THEN 1790
1550 A=STICK(0)+STICK(1)
1560 Q=(A=6ORA=7ORA=8)-(A=2ORA=3ORA=4)
1570 R=(A=8ORA=1ORA=2)-(A=4ORA=5ORA=6)
1580 IF Q=0 THEN B=RND(1)*2 ELSE B=-(Q=-1)
1590 PUTSPRITE3,(120+Q*36,94+R*20),,8+B*2
1600 PUTSPRITE4,(120+Q*36,94+R*20),,9+B*2
1610 IF A<>P OR M(N)<1 THEN 1430
1620 IF F=0 THEN 1690
```

ウーくんっ! (その1)

ウーくんのソフト屋さん

ゲームが開始すると、ウーくんが木の陰からピョコリと現れます。ちょろっ出ることもあれば、ドバーンと出ることもあります。すかさずキーを押しましょう。

ウーくんを見つけると、ウーくんはビックリし、得点が入ります。ゲームは約1分間続きますが、鬼の交代はないので、何度もウーくんを見つけてください。

以前の「ウーくんのソフト屋さん」のものと比べると、リストがとても長くなっていますが、この半分以上はウーくんやミンミンちゃんのキャラクタの絵のデータです。ちゃんとしたウーくんとミンミンちゃんを登場させるために、グラフィックスにもたっぷり凝りましたから、たっぷり楽しんでくださいね。

```
1630 PLAY"O3C64R64C64R64C64"
1640 PUTSPRITE9,(X(P)+LR*16,Y(P)),,4
1650 PUTSPRITE10,(X(P)+LR*16,Y(P)),,5
1660 PUTSPRITE11,(X(P)+LR*16,Y(P)+16),,6
1670 PUTSPRITE12,(X(P)+LR*16,Y(P)+16),,7
1680 M=M+1:GOTO 1760
1690 PLAY"O5C8R16E12"
1700 FORI=5TO8:PUTSPRITEI,(0,217):NEXT
1710 PUTSPRITE13,(X(P)+LR*16,Y(P)),,12
1720 PUTSPRITE14,(X(P)+LR*16,Y(P)),,13
1730 PUTSPRITE15,(X(P)+LR*16,Y(P)+16),,14
1740 PUTSPRITE16,(X(P)+LR*16,Y(P)+16),,15
1750 H=H+1
1760 IF PLAY(0) THEN 1760
1770 COLOR15,12:RX=8:RY=196:PRESET(RX,RY):PRINT#1,USING"Hit:##
Miss:##";H;M
1780 COLOR15,12:RX=8:RY=204:PRESET(RX,RY):PRINT#1,USING"Score:###";
H*10-M*20
1790 FORI=5TO16:PUTSPRITEI,(0,217):NEXT
1800 C=C+1:IF C=40 THEN C=0:D=0
1810 GOTO 1410
1820 '--- Game Over
1830 INTERVALOFF
1840 PUTSPRITE3,(120,86),,8
1850 PUTSPRITE4,(120,86),,9
1860 PUTSPRITE5,(0,216)
1870 COLOR15,12:PRESET(92,102),,OR:PRINT#1,"Time Up!!"
1880 COLOR15,12:PRESET(92,112),,OR:PRINT#1,USING"Score:###";H*10-
M*20
1890 PLAY"O4L16GFEDGFEDCDEFG8"
1900 IF STRIG(0)=0 THEN 1900
1910 GOTO1170
1920 '--- Sub Routine
1930 T=T-1
1940 IF T=-1 THEN RETURN 1830
1950 PUTSPRITE2,(T*2+128,200),15,19
1960 PRESET(RX,RY)
1970 RETURN
1980 '--- Sprite Pattern Data
1990 '0
2000 DATA 08,1C,16,23,23,40,4C,5E
2010 DATA 82,B7,8F,4F,27,1F,1F,20
2020 DATA 10,38,58,8C,8C,06,36,7A
2030 DATA 43,ED,F3,F6,E4,F8,FC,64
2040 '1
2050 DATA 00,08,0C,1E,1F,3F,33,2D
```

ウーくんっ!（その2）

木の陰からは女の子も出てきます。この子を指してしまうと、20点もマイナスされてしまうので、間違えないように注意深くキーを押しましょう。

約1分が経過すると、見つけたウーくんの数と、ミスをした数から、得点がはじき出され、中央に表示されます。今回は140点でした。200点を超えたらエライですよ！

```
2060 DATA 7D,78,77,35,1D,07,0F,1F
2070 DATA 00,10,30,78,F8,FC,CC,B4
2080 DATA BE,1E,EE,AC,B8,E0,F0,F8
2090 '2
2100 DATA 20,40,50,30,10,10,13,1E
2110 DATA 1C,18,00,00,00,00,00,00
2120 DATA 64,06,06,0C,18,18,B8,78
2130 DATA 38,10,00,00,00,00,00,00
2140 '3
2150 DATA 1F,3F,3F,1F,0F,0F,0E,0C
2160 DATA 08,00,00,00,00,00,00,00
2170 DATA F8,FC,FC,F8,F0,F0,70,30
2180 DATA 10,00,00,00,00,00,00,00
2190 '4
2200 DATA 0F,3F,7F,7F,FF,C0,CC,DE
2210 DATA DE,C0,F0,E3,70,7E,76,29
2220 DATA F0,FC,FE,FE,FF,07,37,7F
2230 DATA 7F,07,0F,CF,3E,7E,6C,94
2240 '5
2250 DATA 00,00,00,12,12,3F,33,29
2260 DATA 29,3F,1F,1D,0F,03,0F,1F
2270 DATA 00,00,00,48,48,FC,CC,94
2280 DATA 94,FC,F8,B8,F0,C0,F0,F8
2290 '6
2300 DATA 55,55,35,15,15,35,1F,05
2310 DATA 07,07,05,07,00,00,00,00
2320 DATA 56,56,54,58,58,54,F8,A0
2330 DATA E0,E0,A0,E0,00,00,00,00
2340 '7
2350 DATA 3F,3F,1F,0F,0F,1F,00,02
2360 DATA 02,02,02,00,00,00,00,00
2370 DATA FC,FC,F8,F0,F0,F8,00,40
2380 DATA 40,40,40,00,00,00,00,00
2390 '8
2400 DATA 00,00,05,12,0C,04,1C,04
2410 DATA 03,01,1E,3D,1F,00,00,00
2420 DATA 00,00,E0,10,1E,47,1E,18
2430 DATA F0,E0,C0,80,00,00,00,00
2440 '9
2450 DATA 00,00,00,01,03,03,03,03
2460 DATA 01,00,01,1F,00,00,00,00
2470 DATA 00,00,00,E0,F0,BE,F0,F0
2480 DATA E0,80,80,00,00,00,00,00
2490 '10
2500 DATA 00,00,07,08,70,B2,70,10
2510 DATA 0B,06,03,01,00,00,00,00
2520 DATA 00,00,A0,48,70,60,78,E0
2530 DATA C0,80,F8,E4,F8,00,00,00
2540 '11
2550 DATA 00,00,00,07,0F,7D,0F,0F
```

ウーくん、ミンミンちゃん達が活躍する、ほのぼのマンガです。かつてのMSXマガジンでは、本コーナー「ウーくんのソフト屋さん」が大好評でした。今回、とっても久しぶりに、ウーくんが帰ってきたのです！（やほう）

ウーくん
心優しいブタ。普通のブタと違うのはネッカチーフを巻いているところ。とてもオシャレで、その日の気分によって柄を使い分けている。バリやハワイといった南の島が好きだが、ただでさえ脂性なので日焼け止めクリームは油分の少ないものを愛用しているとか。どうやらこれまでずっと寝ていたらしい。

ミンミンちゃん
中国生まれのミミズ。大陸にいる両親のことを考えてついつい泣いてしまう。カフェオレカップでの泥あびが好き。カフェオレカップそのものへのこだわりもあり、陶器や磁器のいい品を探しに京都まで出かけていってしまうほど。昔はゲーマー志望だったけど、今回ゲームプログラマとして活躍していることが発覚。

```
2560 DATA 07,01,01,00,00,00,00,00
2570 DATA 00,00,00,80,C0,C0,C0,00
2580 DATA 80,00,80,F8,00,00,00,00
2590 '12
2600 DATA 08,1C,1E,3F,3F,7F,7F,7D
2610 DATA FB,FF,F8,7A,3A,1F,10,3F
2620 DATA 10,38,78,FC,FC,FE,FE,FA
2630 DATA F7,FF,1F,5E,5C,F8,08,9C
2640 '13
2650 DATA 00,08,0C,02,1D,3E,36,3E
2660 DATA 5D,60,77,35,1D,07,0F,1F
2670 DATA 00,10,30,40,B8,7C,6C,7C
2680 DATA BA,06,EE,AC,B8,E0,F0,F8
2690 '14
2700 DATA 3F,7F,7F,BF,5F,1F,1D,1A
2710 DATA 14,08,00,00,00,00,00,00
2720 DATA CC,CE,FE,FD,EA,E8,E8,68
2730 DATA 28,10,00,00,00,00,00,00
2740 '15
2750 DATA 1F,3F,2F,4F,0F,0F,0E,0C
2760 DATA 08,00,00,00,00,00,00,00
2770 DATA F8,FC,F4,F2,F0,F0,70,30
2780 DATA 10,00,00,00,00,00,00,00
2790 '--- Sprite Color Data
2800 DATA 42,42,42,42,42,42,42,42
2810 DATA 42,42,42,42,42,42,42,42
2820 DATA 42,42,42,42,42,42,42,42
2830 DATA 42,46,42,42,42,42,44,44
2840 DATA 44,42,42,42,42,42,42,42
2850 DATA 42,42,42,42,42,42,42,42
2860 DATA 42,42,42,42,42,42,42,42
2870 DATA 42,42,46,42,42,42,4A,4A
2880 DATA 4A,4A,4A,4A,4A,4A,42,42
2890 DATA 42,42,42,42,42,42,42,42
2900 DATA 4E,4E,4E,4E,4E,4E,4E,4E
2910 DATA 4E,4E,4E,4E,4E,4E,4E,4E
2920 DATA 4E,4E,4E,4E,4E,4E,4E,4E
2930 DATA 4E,4E,4E,4E,4E,4E,4E,4E
2940 '--- Palette Data
2950 DATA 0,0,0,0,0,0,7,6,5,7,5,5
2960 DATA 7,6,5,1,5,7,7,6,5,3,1,1
2970 DATA 2,6,7,7,6,5,6,6,0,5,4,0
2980 DATA 0,3,0,1,4,1,5,5,5,7,7,7
2990 '--- Tree Position Data
3000 DATA 120,36,184,44,220,96
3010 DATA 184,148,120,156,56,148
3020 DATA 20,96,56,44
3030 '--- Movement Data
```

ウーくんのソフト屋さん

ウーくんっ！（その3）

目指せクリエイター Part1

MSXお絵描き講座

グラフサウルスver2.0 CG職人への道

グラフサウルスver2.0は「クインプル」や「ファミクルパロディック」で有名なBIT²が発売していたグラフィックツールだ。複数のグラフィックスモードに対応しているツールとしては、MSX史上最強といっても過言ではない。ゲームメーカーが、ゲーム制作でグラフサウルスを使っていたといった話はいくらでもあるのだ。そんなプロのゲームデザイナーも愛用のグラフサウルスver2.0を本誌付属CD-ROMに収録することができた。これを使ってCGしてくれ〜っ！

付属CD-ROMに収録

グラフサウルスver2.0の基本操作だ

Windowsと比べてしまうと、グラフサウルスver2.0の操作には独特なものがある。そこで、起動／終了、メニューの出し方、画像データのロードとセーブなど、基本操作をおさらいしておこう。

グラフサウルスver2.0の起動

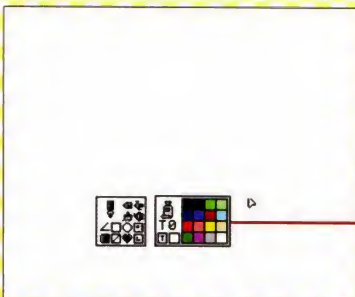
スタートメニューから「グラフサウルス」を起動するとタイトル画面が表示される。[スペース]キーを押すと、グラフサウルスがマウスで動作するモードになる。ところが、このままでは手元のマウスを動かしても、グラフサウルスのマウスカーソルは動いてくれない。本誌付属のグラフサウルスは、WindowsとMSXPLAYerのマウスカーソルを切り替えて使うようになっているからだ。

そこで押してほしいのは[F12]キー。これで、Windowsのマウスカーソルが消え、グラフサウルスのマウスカーソルが動かせるようになる。Windowsのマウスカーソルを再び動かしたいときには、再度[F12]キーを押せばよい。

画面モードの選択

グラフサウルスのマウスカーソルが動くようになったところで、先ほどから表示されているタイトル画面をクリックする。すると、描画する画面モード（SCREENモード）の選択メニューが表示される（図1）。グラフサウルスは、SCREEN5/7/8/12に対応しているので、これを選ぶ必要があるというわけだ。今回はSCREEN5をクリックして、そこでイラストを描くことにしよう。もっとも他の画面モードでも操作は同じ感覚でできるので、気が向いたら他を使ってみるといい。

マウス操作の基本

SCREEN5を左クリックすると真白の画面にアイコンとパレットが表示される（図2）この状態でCGイラストを描いていくんだ。マウスボタンの使い方はMSXもWindowsもだいたい同じ。<左クリック>で「決定」、<右クリック>で「キャンセル」だ。ただし<右ダブルクリック>という操作があって、これを行うとメインウィンドウ（図3）が表示される。グラフサウルスでは、このメインウィンドウのアイコンで使う機能を決定して描画をするのが基本的な使い方となっている。

グラフサウルスver2.0の機能

グラフサウルスにはたくさんの機能がある。図4の落書きサンプルは、その中のライン（線）、サークル（円）、ボックス（四角）、ペイント（塗りつぶし）、文字出力などの機能で描画したもの。その他にもいろいろな機能があるが、それらは次の「実録！グラフサウルスver2.0で絵を描こう」のコーナーで覚えてほしい。なお、地味だけど役立つ機能として、右クリックで使えるスポイト機能がある。この操作で、マウスカーソル位置の色情報を吸い取って次の色塗りに活かすことができるのだ。すでに使用されている色を、即使いまわせるので作業がはかどるぞ。

ロードとセーブ

CGイラストを描いたら、データのセーブ（保存）をしよう。また、一度保存したデータはロード（読み出し）すれば表示できる。これらの操作は、メインウィンドウのディスクのアイコンをクリックして、次に表示されるディスクウィンドウで行う（図5）。このウィンドウを閉じるには編集画面を右クリックだ。

描いた画像データをセーブしたいときには、[SAVE]をクリックする。するとファイル名（最大8文字。拡張子はスクリーンモードによって固定されている）が尋ねられるから、入力して[リターン]キーを押すか［RT］をクリックしよう。後は［OK］ボタンを押せば保存される。このとき、すでに存在しているファイルと同じ名前で保存すると上書きされてしまうので注意だ。

画像データをロードしたいときには、今度は［LOAD］をクリックする。すると画像ファイルの一覧が表示されるから、ファイルを選択して［OK］をクリックしよう。これでファイルが読み込まれて画像が表示される。なお、ロードすると作成中のデータが消えてしまうのでこちらも注意だ。

なお、MSXPLAYer版のグラフサウルスは、ディスクA（システムディスク）とB（ユーティリティディスク）にも書き込めるようになっているが、自分で描いた画像を保存したいときには、あらかじめMSXPLAYerのディスクを、AとB以外に切り替えること。それからディスクウィンドウで保存をしよう（図6）。

MSXミニギャラリー

気合いを入れて描かれたMSXグラフィックスの数々。がんばればこんな絵を描くことも可能だ。これらのCGイラストは、いろいろなテクニックを用いて描かれているので参考にしよう。

グラフサウルスVer2.1のサンプルからファンキーKの作品。スクリーン7の画像で16色しか使えないがタイルパターンを駆使して表現している。

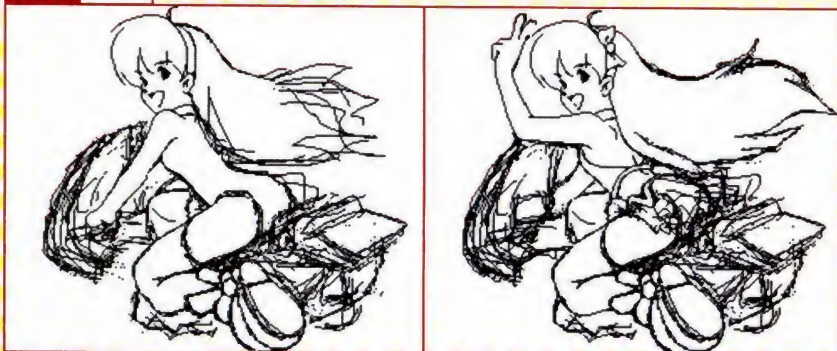
オリジナルでまかろんの作品。今回の教材と同じスクリーン5で16色の大半を使用した緑色のグラデーションが非常に美しい。ドット修正の賜物である。

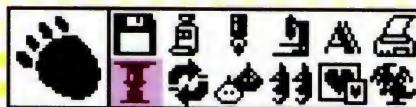
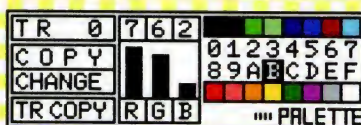
うえのまさひろ作で教材と同じ女の子、魔女っ子Mちゃん。背景の画像は256色使えるスクリーン8だからこその芸当。スクリーン12でも見てみたい？

グラフサウルスのディスクBには画面モードごとのサンプル画像がいくつか入っているのでロードしてみよう。そのクオリティには、ちょっとビックリするはず。なお、画像を読み込んだ後は必ずディスクAに戻しておくこと。こうしておかないとディスクウィンドウを消すことができないのだ。

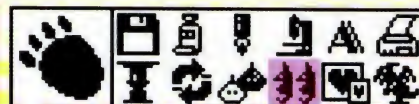
実録！ グラフサウルスver2.0で絵を描こう

基本操作がわかったところで、実際にグラフサウルスver2.0でCGイラストを描く過程を紹介しよう。自分でCGイラストを描く際の参考にしてもらいたい。

STEP 1 ラフを描く

ラフの線を書いていきます。ラフなので線のサイズなどは何でもいいのですが、後の修正が少しでも楽な細めのペンを選びます。これでゴリゴリとイメージを固めながら適当にいろいろ描いていきます。鉛筆での下書きの段階ですね。

STEP 3 デッサンの確認

左右反転をさせてデッサンを確認します。左右反転するには、リバースのアイコンを左クリックして横矢印を左クリック。画面全体をドラッグする感じで全画面を選択して、再度左クリックします。この後は、楽しい楽しい色塗りに入ります。

STEP 4 パレットの調節

まず肌色を塗りましょう。初期の色には、いい色がありませんので、パレットの変更を行います。SCREEN 5で使用できる色は16色なので、色の割り振りもこの段階でだいたい決めてしまいます。作成した肌色で、肌の部分を塗ってしまいます。

STEP 7 つなぎ目の修正

途中まで作業を続けていたところ、どうも胴体が少々長いような気がしたので修正を行います。コピーを利用して、お腹から下の領域をすこ～し上に移動させます。つなぎ目の微調整をして、このように修正してみました。

STEP 2 ラインの整理

ある程度の形になり、まとまってきたら、ルーペウインドウやイレース機能を使用してラインの整理／修正に入ります。この段階で手間をかけておくほど後の作業が楽になるので、気合いを入れて作業をします。

グラフサウルスver2.0のマニュアルを本誌付属CD-ROMに「GRAPH_SAURUS.PDF」（PDFファイル）として用意したので、詳しい使い方はそちらをご覧になっていただきたい。

STEP 5 色塗り

次に髪の色を塗ります。…ってひどいことになりました。どうやらラインの一部が切れていたようですね。こういうときは、あわてずにアンドゥ(やり直し)をして、塗る直前の状態に戻します。アンドゥできないときは、他の色でもう一度塗るのも手です。

STEP 6 塗りなおし

線の切れている一部をルーペウインドウで拡大して修正し、もう一度塗りなおします。他の個所の塗りも同様にやっていきます。STEP2でラインの修正をしっかりやっておけば、塗り自体はそんなに手間ではないでしょう。

STEP 8 背景との重ね合わせ

ここからラストスパート。ルーペウインドウでの作業が中心となります。髪の毛の光の当たっている部分などを修正していきます。修正が終わったらセーブ。これを忘れて泣いたことがあります。最後に、別に作成した背景と組み合わせて完成となります。

完成！

Close-up Games

どんなに名作なゲームでも先に進めなきゃオモシロくない。昔は近所に仲間もいたし、リアルタイムで雑誌もあったし、頭脳や手足はゲーマーに最適化されていたしで、当時はゆうゆうとクリアしていただろうが、10年以上経過した今ではトンデモ激ムズ。そんな「昔はクリアできたんだけどなぁ」なんて肩を落としそうなキミに、Mマガがチョイスしたゲーム5本についての情報を大公開だ。これで記憶をビキーンと呼び覚ませ！

ザ・キャッスル キャッスルエクセレント

MSXの「あの」名ゲームを収録。もちろんハードディスクにセーブもできる。しかも全マップ公開ときた！

TEXT／新保剛平
アスキーにてザ・キャッスル、キャッスルエクセレントのプロデュースをしていた人物。キャッスルへの思い入れも人一倍だ。

まったりしてても ゲームは骨太

「恋する王子の100の冒険」そんなキャッチフレーズがついていたゲームだ。2頭身のかわいいキャラクタ、のんびりしたBGM、まったりしたゲーム進行と、それだけ聞くと脱力しそうだが、実際にやってみるとこれが非常にエキサイティングで奥が深いのだ。

舞台は10×10＝100の部屋から成るグロッケン城。このどこかにマルガリータ姫が捕らえられている。そして、姫を助けるために主人公たるラファエル王子が城に挑むというストーリーだ。各部屋にはさまざまな仕掛けがあるし、敵キャラもウヨウヨ。部屋から部屋へ移動するには、扉と同じ色の鍵をどこかで調達しておかねばならないのだが、近くの部屋で見つかるとは限らない。つまり、城全体が1つのステージとして構成されているようなものなのだ。

そんなキャッスルを攻略する基本的なポイントは2つある。1つ目はラファエル王子のちょっと変わったジャンプ能力。とにかく滞空時間が長いし、移動距離もかなりのもの。それゆえ、一見登れそうにない高い場所や、渡れそうにない遠い場所へのジャンプが届いたりする。このあたりの距離感は何度もトライしてつかんでおこう。さらに、ジャンプ中は方向転換ができたりもする。このワザを用いると、左に飛んで右に戻り、真上の段に登ることもできるのだ。

2つ目のポイントは命の扱いだ。ラファエル王子は最初、命を5つ持っているが、薬ビンを取れば命がどんどん増える。逆に、敵キャラに触れたりエレベータに挟まれると1つ減ってしまう。命は途中セーブする対価としても使えるから、下手に命を減らすぐらいならば、ときにはケチケチせずに途中セーブに使いたい。

ほかにも、覚えておきたいことはたくさんあるが、それらは実践で理解していくとして、ここでは特に難解なシーンを中心に攻略方法を紹介していくぞ。

キャッスルと同じシステムで、さらにパズルを難解にして、アクションを難しくしたのがキャッスルエクセレントだ。p.112から続く、お城のマップを見ただけでもクラクラしてくるだろう。
ここではエクセレントの中心となる2つの大仕掛けである脱出ポットと、バリアによるブロック交差について解説しておこう。

ここはエクセレントの最後の難関。姫の部屋（下）とその直前の部屋（上）だ。複雑に一方通行が仕掛けられているのがわかるだろう。じつはゲーム中盤に姫の部屋は見ることができる。ただし計算されつくした鍵の配置によって、すぐに助けることができないのだ！。

ザ・キャッスル攻略編

ザ・キャッスル　序盤戦の大関門　207号室

ザ・キャスルを進めるうえで、序盤戦最後の難関となる部屋を解説しよう。

それは地上2階の左から7部屋目にある。別にたいした部屋に見えない。わざわざ解説する必要があるかどうか疑問に思う人も多いだろう。だがこの部屋が強烈に難しいのだ。空飛ぶブロックと、横針地獄になっている。

この画面を見たらまずセーブ。絶対にあなどってはいけない。ちなみに直前の部屋には不死身光線の発生装置が用意されているが、これはオトリ。最高のタイミングで不死身光線を浴びても、この部屋の針には間に合わないのだ。右下から入ってきて、針を飛び越し、左に脱出する。水色の鍵が2つ必要だ！

一見簡単そうな面なんだけど、やってみるとどうしても中央の横向き針にグサリと刺さっちゃうのだ。

ブロックが左に飛んで行くときに、この位置に飛び込む。落ちた場所から右に半歩がミソ。

下のブロックに注目。右に飛んできたときに、小さくジャンプしてその場で乗る。

下のブロックの上から左に大ジャンプ！　ここからスペースキーは押しっぱなしだ。

上の針を越えて下降するときに、上のブロックが足元にきている。そのままジャンプ。

無事渡り終わった王子。下の緑のキーは、半歩のり出してからジャンプするのがポイントだ。

キャッスルエクセレント攻略編

バリアで進める部屋が交差する？　307号室

エクセレントでは、いくつかの部屋が見えないブロックを構成している。ブロック内は、自由に行き来できるのだが、隣のブロックの部屋に行くには鍵や、空気タンクが必要になるわけだ。

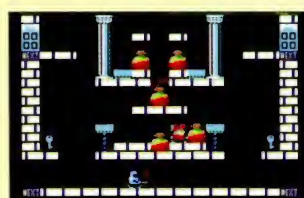
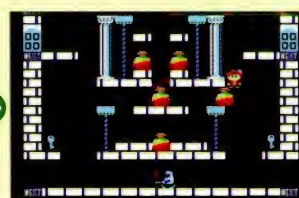
バリア交差は、そんなブロックとブロックを交差させる仕掛け。左下からエレベータで上がってきた王子は右上に抜けるしかない。反対に右下から上がってきた王子は、左上に抜けるしかない。バリアクロスは307号室、808号室などにあるぞ。

下の部屋からエレベータで上がってきた王子。

扉はいっぱいあるけれど……。

進めるのは右上の扉だけ。これがバリア交差だ。

脱出ポットの数しか通り抜けできない　507号室

脱出ポットの典型例が507号室と609号室だ。右の507号室の連続画面を見てほしい。下から入ってきて上の扉に行くには、必ず壺を潰す必要がある。使える壺の数は2つ。つまりこの部屋は最大2回しか通り抜けできないのだ。

右上の扉はジャンプしても届かない。

壺をエレベータに載せて、その上に立つ。

壺が潰れる瞬間ジャンプ。これで脱出！

ゴンドラのこんな使い方って！　802号室

次に紹介する攻略法は、キャッスルの最終章、802号に現れるちょっとひねったゴンドラパズルだ。

左から入って右の扉から脱出する。でも扉には背が届かない。レンガを扉の前まで押していくことになるのだが、さてさてその方法は……。

レンガの移動の手順と騎士をやっつける手順、ゴンドラの上下、王子の移動する空間など、これらすべてを考慮しなければならないから大変だ。

右上の扉の前に、レンガを移動するのが最終目標。

右下の命の薬をとったら、レンガはこの位置に。

邪魔な上の2人の騎士は、レンガを押してやっつけてしまおう。

上のレンガをゴンドラを通して落とす。

ゴンドラを下ろして、下の騎士をブロックでボシュッ。

この形にレンガを積む。これがポイントだ。

レンガを半分だけ押すんだ。でも押し過ぎに注意して。

ゴンドラを持ち上げると、ほら！レンガはこのとおり。

はい。これで脱出完了！

部屋番号は全マップで確認しよう

攻略解説で使用しているザ・キャッスルとキャッスルエクセレントの部屋番号は、全マップの左から右へ1、2、3……とナンバリングし、上下は地下からB2、B1、1F、2Fと通常の建物と同様に階数で表示してある。たとえば207号室なら、2階の左から数えて7番目の部屋という具合だ。全マップには、わかりやすく階数と番号が振ってあるので、参考にしてほしい。

裏技を伝授しよう！

1 スピードコントロールはキーボードで

キャッスルのテンポが遅いと思う人は多いだろう。特にエレベータを待つときなんかはイライラだ。そんなときは[CTRL]キーを押そう。これで中速モード。さらに[CTRL]キーと[GRAPH]キーを同時に押すと高速モードになる。

2 ときには命を犠牲にすることも必要

大事な壺やブロックをエレベータなどで潰してしまった場合は、[F1]キーでLifeを1つ犠牲にすれば復活するぞ。ただし、これは部屋を出る前までだ。いったん部屋を出てしまうと、このワザを使ってもブロックは復活しなくなってしまう。そのときはセーブした場面からやり直すしかない…。

3 セーブファイルを使いこなせ！

キャッスルには「詰み」と言われる状態が起こる。大事なブロックを潰してしまったり、鍵を取り忘れたりして、進むことも戻ることもできなくなった状態だ。

ザ・キャッスル攻略編

「鯉の滝登り」ならぬ「王子の壺登り」？　403号室

この部屋は、入った瞬間に勝負が始まる。上から樽、ローソク、壺がどんどん落ちてくる。その間をすり抜けて、三角飛びを繰り返しながら、登らなければならない。順番に見ていこう。

右から入って、左に脱出。でも部屋に入ったとたんに樽が降ってくるぞ。

樽につぶされないよう大急ぎで進め!!

柱まで行ったら、すぐに振り向きながらジャ～ンプ。急がないと落ちてくるローソクにつぶされるぞ。

いったん樽に着地してその上からジャンプ。

落ちてきた壺の上を目指して飛ぶ。

壺に着地するが、うかうかしている時間はないぞ。

そのまま右上を目指してジャンプ！

空中で方向を変えてもう一度壺の上を目指せ。

最後の壺が落ちてくる前に、大急ぎで隙間に入る。

これで壺登りの完成だ。左の空飛ぶブロックへ進もう！

あの攻略本をCD-ROMに収録！

キャッスルエクセレントと同時に発売された「キャッスルエクセレントスーパーヒントブック」を本誌のCD-ROMに収録している。

PDF形式になっているので、読むにはアクロバットリーダが必要だ。アクロバットリーダは、アドビのWebページからダウンロードしてほしい。URLはhttp://www.adobe.co.jp/だ。

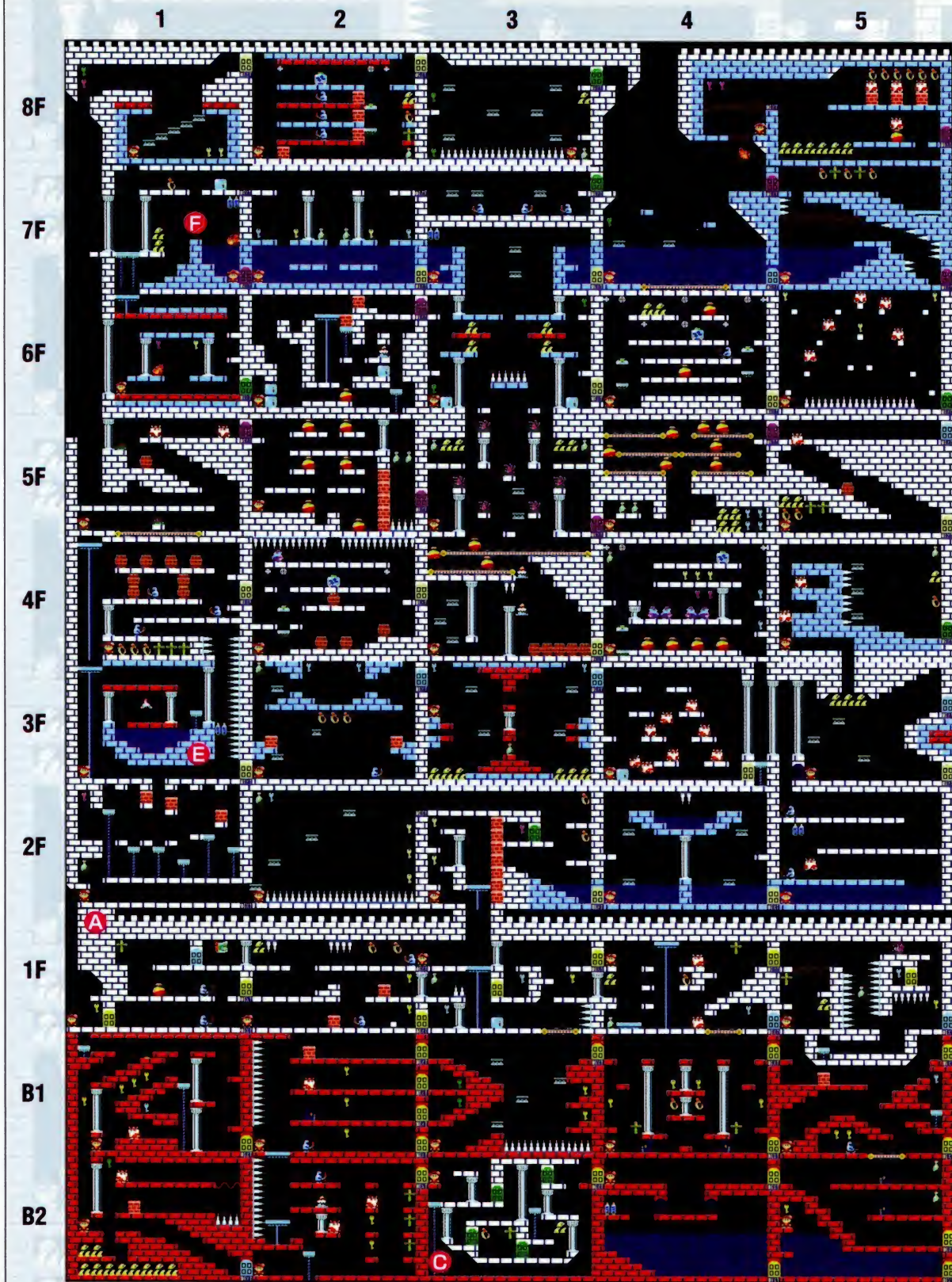
ROM版のキャッスルでは、カセットテープに途中シーンがセーブできた。そしてカセットテープを取り替えれば好きなどのシーンにも戻ることができたのだ。しかし、本誌のMSXPLAYer版では、これをファイルとしてディスクにセーブするように改造してあるため、連続してセーブすると、同じファイルに上書きされてしまい、最後のシーンしか残らないのだ。

そこで裏ワザを教えてしまおう。それはセーブしたファイルの名前を手作業で変える方法だ。右画面を見てほしい。セーブしたシーンデータはsavedata.cas※というファイルに保存される。したがって、これをたとえば「savedata1.cas」と変更しておけば、複数のシーンをセーブできる。そして、あのシーンに戻りたいと思ったらセーブしたファイルの名前をsavedata.casに戻してキャッスルでロードすればよい。

ここで注意なのだが。この技を使うときには、必ずMSXPLAYerをいったん終了しておくこと。起動されている状態では、正しくシーンのセーブやロードが行われない可能性がある。めんどうでも終了してから作業してほしい。

この技を紹介するのも、はっきり言って、キャッスルの攻略が難しいからだ。何が何でも完全攻略したいときには、こうした知恵も役立ててみてほしい。

※通常は次のフォルダの中にある。【ザ・キャッスル】C:¥Program Files¥ASCII¥MSX MAGAZINE 2003 ゴキゲン8bitゲーム（スキン名）¥msx¥img¥ascii¥thecastle 【キャッスルエクセレント】C:¥Program Files¥ASCII¥MSX MAGAZINE 2003 ゴキゲン8bitゲーム（スキン名）¥msx¥img¥ascii¥castleex

これがザ・キャッスル全マップだ!!

Ⓐここからゲームが開始。この部屋には地図が見えてる。でも取るには水色の鍵が必要だ。

Ⓑこの水色の扉を開けて、地下に入っていく。どんな世界が待っているのか。

Ⓒこの部屋を通り抜けるには、緑の鍵が4つも必要だ。この後水中探険が待ってるぞ。

Ⓓ妖精発見。助けると赤い鍵をくれる。

Ⓔ2人目の妖精発見。でも空気タンクがないと助けにいけないよぉ。

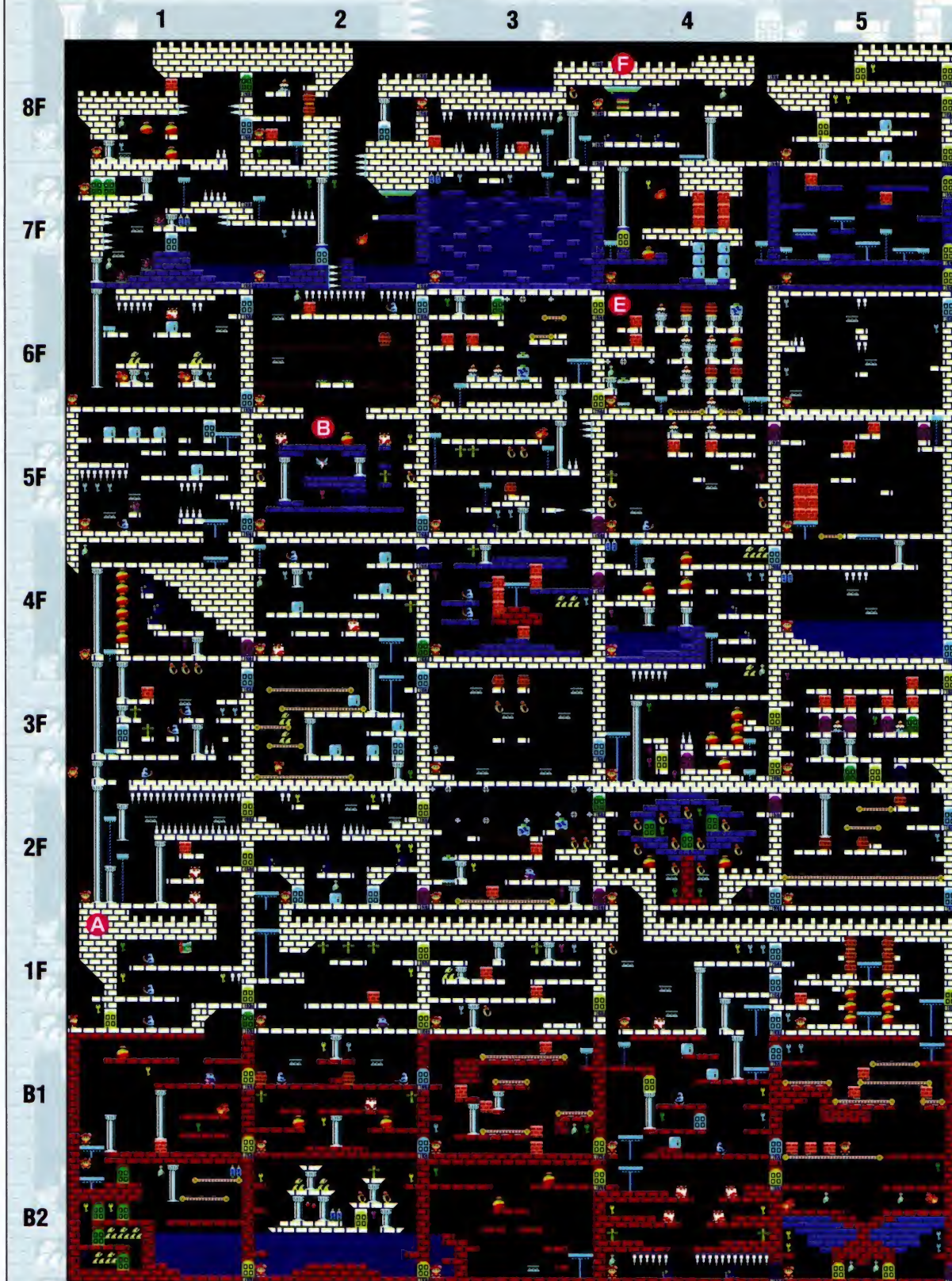
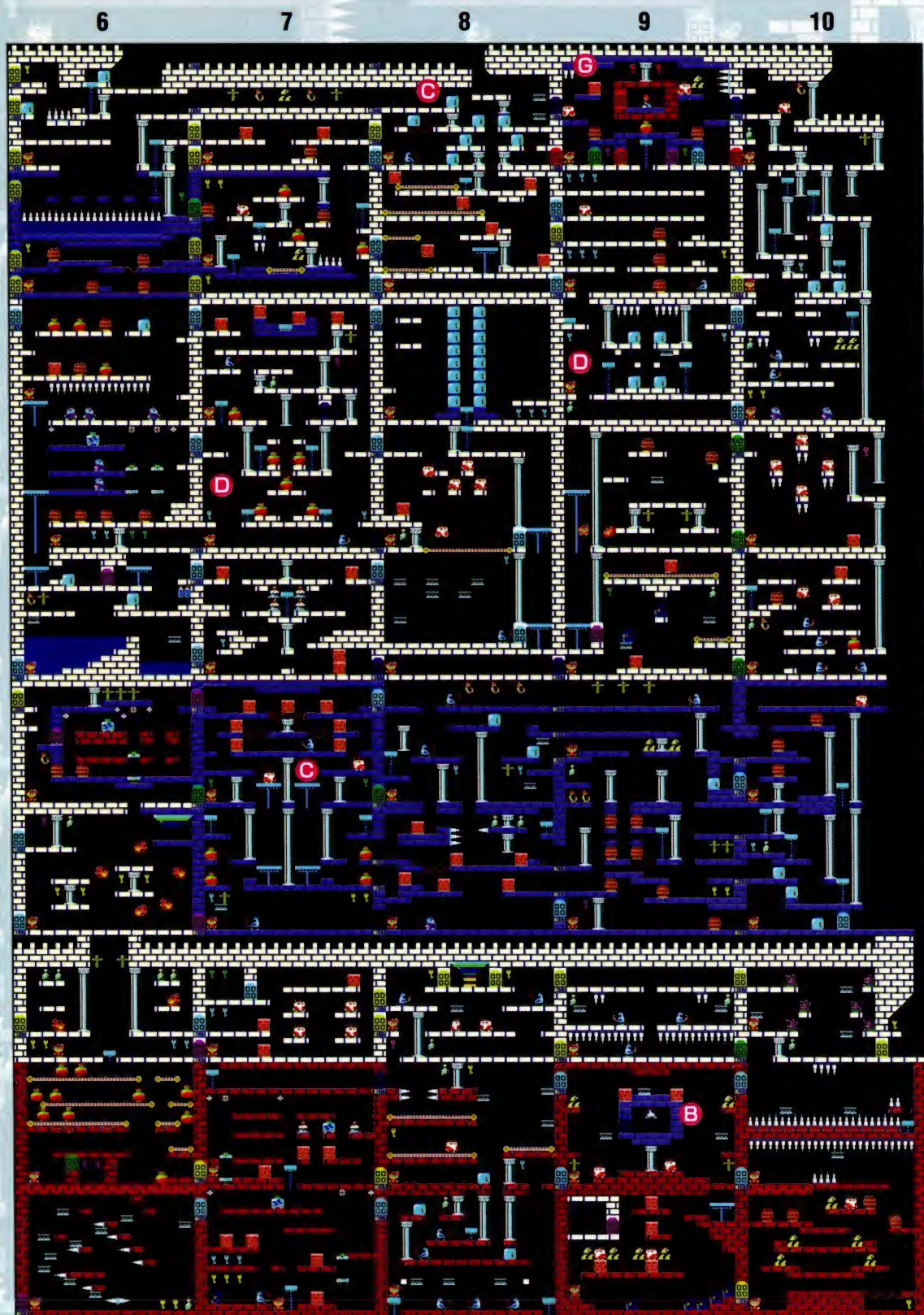
Ⓕ雨水がたまったのか、こんなとこでも水中探険。空飛ぶブロックからは絶対落ちないように。

Ⓖ謎の物体チェリーを発見。でも、このチェリーを取ると、どこまで落ちてしまうのやら。

Ⓗこれが憧れのマルガリータ姫の部屋。姫様、お慕いしております。

これがキャッスルエクセレント全マップだ!!

ザ・キャッスル／キャッスルエクセレント

6 7 8 9 10

【注意!】アルファベットで指定する箇所は2つの部屋を指している場合もある。

Ⓐスタートは前作と同じ。でもマップを取るには上の部屋から落ちてくる必要がある。

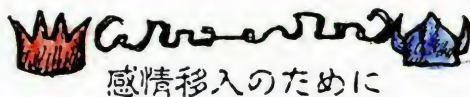
Ⓑ妖精も2匹いる。赤い鍵をくれるぞ。

Ⓒこれがバリア交差の部屋。進める部屋の交差点になっている。よーくつながりを読みこなそう。

Ⓓ脱出ポットの部屋。通過できる回数は限られている。よーく気をつけて使う必要があるぞ。

Ⓔキャッスル全体を通して、一番面倒なパズル部屋。でも完成した姿はちょっと美しかったり。

Ⓕ不死身光線を浴びてエレベータで屋上に突き抜けろ！そのまま屋根を伝って、姫の部屋へ！

Ⓖ姫の部屋。右から左から何回も入り直す必要あり。しかもⒹの脱出ポットと密接に絡んでいるぞ。

ボコスカウォーズ

ボコスカウォーズの作者が熱く語る！ これぞ、ボコスカウォーズの歴史絵巻

TEXT／ラショウ
ボコスカウォーズの作者であり、ソフトハウス「イタチョコシステム」の総帥。今回直々に筆をとり、本文のみならずイラストまで手がけている。

感情移入のために

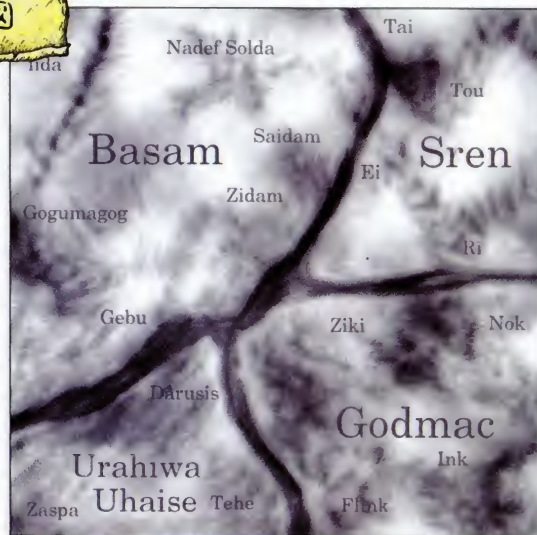
はじめに ～作者より～

今でこそ謎のソフトハウス「イタチョコシステム」の総帥として、そして奇跡の天才ゲームデザイナーとして不動の地位な私であるが（しつこいな宣伝）、ありがたくもそんな私が直々にボコスカウォーズのプレイの仕方を、いやさ、MSXゲームというものの捉え方を、果てはMSXの思想！理念！……までを掘り下げて行こうと思うものである。ちょっと言い過ぎだな……。多分うまくいかないです……。

最近になってゲームは構造がシンプルな程、より想像力が膨らむという事にみんなが気づき始めた。そこからレトロゲームの復権が起こったのかも知れないね。ならばそのゲーム解説は、想像力を誇大妄想的に膨らましていくのが正しいのだろう。正しいに決まってる。さあ、みなさん。ご一緒にボコスカウォーズ世界を歴史を楽しく構築してみよう！ええ、どうせなら極端に！ もちろんその中にゲームの攻略法をも盛り込んでいくという挑戦をするので、その意気込みだけは評価して下さい……。だけは……。

ボコスカ世界のはじまりはじまり！

ボコスカウォーズとは――スレン歴0024年2月に始まり、0026年11月に終わったスレン王決死の領土奪回作戦。その作戦名であったが、バサム帝国側の様々な対抗も総じて現在ではボコスカウォーズと呼ぶ。

地図

イタランドについて

ランドといっても島ではなく、大陸であったらしい。古代よりこの地を治めていた、ウラヒワ王朝、ウハイス王朝を南の砂漠に追いやった形で騎士団形態のスレン王国が誕生した。スレンは善政を施き、民心を集めていたが、その数年後、皇帝オゴレスが西方より現れ、西に強大なバサム帝国を建国、スレン王国を圧迫していったのであった。東からは蛮族の侵入も常にあったようである。存亡の危機に至り、遂にスレン王は決戦ボコスカウォーズを決断したのである。

ボコスカウォーズ

ボコスカウォーズ登場人物

皇帝オゴレス(バサム)

力こそすべて。バサム帝国初代皇帝。居城に籠もり、すべては謎に包まれている。スレン王以外には無敵！

スレン王(スレン)

主人公。義理に厚い温情型。24才。年齢的、性格的なものか、剣技のほうは出陣当時はまだあまり冴えていなかったようである。

重騎士クルサ(スレン)

蛮族制圧に辺境に赴いていたがボコスカウォーズの報を聞き、一介の騎士の姿に変装し、紛れて馳せ参じる。

騎士ラクサ(スレン)

スレンに忠誠を誓い、いつまでもどこまでもスレン王に付き従う、まさに騎士の鑑。ああ、献身。

親衛隊長グディノウ(バサム)

バサムの親衛隊長、無敵の魔法鎧をまとう。

兵士長フィ(バサム)

不吉なるバサムの赤い彗星。スレン王を執拗に追う。

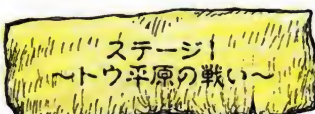
重騎士ダリエイ(バサム)

不敗将軍の異名を持つ。重騎士軍団を率いて旧スレン城跡にてスレンを迎え撃つ。

魔導士イヨー(バサム)

バサム最大の魔導士にして策略家。

収監砦グマゴグ

バサム帝国皇帝オゴレスはスレンの領土を席巻、捕らえた各地の名士を5つの収監砦に幽閉していった。それら捕虜の救出にあたり、スレン騎士ラクサは強固な鉄格子を命を賭した体当たりによってこじあけたという逸話がある。

オゴレスに東の果てトウの地に追い詰められてなお、スレンの気持ちは萎えてはいなかった。

スレンを取り巻く兵達はいずれも新参、弱輩の兵達ばかりになってしまっていた。しかし彼等のどの眼差しにも弱気の色はなく、全てに蒼きスレン魂が映っていた。また、ラクサ騎士団の残存と合流出来たのは幸いであった。この初期布陣を蒼き魂の方陣という。特にMSX版はこの形を正確に再現しているのである。

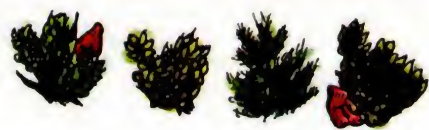
スレンは静かに西を見、出陣を言い放った。ボコスカウォーズの始まりである。

まずスレン軍のすべき事は、これまでスレン国民を苦しめ残虐非道の限りを尽くしていたバサム帝国の侵略軍団兵士長フィを打ち破る事であった。

もちろん既にフィは残存スレン軍の集結を一早く察知し、トウ平原にてこれを迎え撃つ陣を敷く。スレンの地を蹂躙していたフィ軍の恐ろしさはその機動性であった。縦横無尽のフィ軍にスレン王はいきなり悩まされる事になる。

さてボコスカウォーズ。その作戦とは各地の捕虜を救出しつつ、敵軍を急襲、撃破し、あるいはすり抜け、同時に自らと兵士達の戦闘経験をも積んでいこうというものだった。そう、そもそもがそんな無謀な作戦であったのだ。勝算は低かったが、スレン王に選択の余地はもうなかったのかもしれない。誰もがそのシミュレートにおいてその至難を実感することになるだろう……。

ボコスカウォーズ年表

年月	出来事
Stage1	
0024年 2月	スレン、「蒼き魂の方陣」を組む ボコスカウォーズ開戦
5月	トウ平原の戦い
8月	収監砦第一グマゴグ解放
Stage2	
0024年 9月	タイ森林戦始まる
0025年 3月	重騎士クルサ、スレン軍に合流 スレン、聖剣「キキルマンマ」獲得
4月	収監砦第二グマゴグ解放
Stage3	
0025年 5月	スレン軍、東ナデフソルダ峡谷に迷い込む
10月	スレン、ボコスカウォーズ進軍歌作詩
11月	収監砦第三グマゴグ解放
Stage4	
0026年 1月	スレン軍、サイダム解放
2月	バサム軍ダリエイ将軍サイダムを奪還
4月	旧スレン城決戦
6月	サイダム解放、収監砦第四グマゴグ解放
Stage5	
0026年 8月	フィの屈辱(フィ失踪)
9月	収監砦第五グマゴグ解放
11月	スレン軍、バサム城へ！
11月	決着！勝利者は？？

トウ平原の戦いは奇跡的なスレン軍の勝利に終わった。最初の収監砦の撃破も大いに士気を高めるものとなった。だが、バサム帝国の天才的な策略家魔導士イヨーはすでにある対抗策を練っていた。

イヨーはタイ森林にスレン軍を誘い込み、雪辱に燃えるフィ軍を再び差し向けた。そこに釘付けにし、彼等を疲弊させ、じりじりと兵を削っていこうとしたのである。その一方で、東ナデフソルダ峡谷に着々とある「仕掛け」を作っていたのだ……！

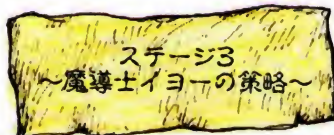
暗い森の中、水を得た魚のように神出鬼没の機動フィ軍。苦戦を強いられていたスレンであった……。その時である。

「我、スレン重騎士クルサなり！」重騎士クルサが遠方エイ戦地より馳せ参じたのだ。

彼は、一振りの剣をスレン王に奉じた。これこそ大木をも切り倒せるという伝説の聖剣「キキルマンマ」であった！　早速、王はその剣を振るい、大木を次々と一撃で切り倒し、道を開いて自軍を導き、フィ軍をかわしていった。

また、重騎士クルサはフィ軍の動きの秘密を看破していた。「フィはスレン王の位置を見ながら作戦の発令を変えていた」のだ。それは実に驚くほど簡単な用兵であった。遂にスレン軍は難敵フィを退け、森の脱出に成功する。

しかし、スレン王は開けた前方の視界を見て驚愕した。結局スレン軍は魔導士イヨーの思うところの東ナデフソルダ峡谷にまんまと導かれていたのである……。

ステージ3
～魔導士イヨーの策略～

東ナデフソルダ峡谷。ウオオォン、オオォン、オォン。乾いた土地の岩が声を幾重にも反響させていく。

そう、この地は、妖術使いにとってうってつけだったのだ。

突然、岩の陰から立ち上る敵兵！　いや、幻……なのか？　実は、その敵兵士はイヨー配下の魔導士達の妖術で生み出された幻戦士であった……。スレン軍の若い兵士達は、ある者はそれに惑い狂乱し、ある者は同士討ちをしてしまう有り様だった。このままではせっかくの収監砦の成果を失ってしまう……。

すべての兵は疲れ、助けを求めるように希望の星であるスレンの王を見た。

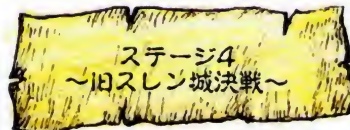
若きスレンの王は、いや、彼は明らかにこの戦で数段の成長を遂げていた。スレン王は、目を覚ませとばかり、朗々と歌を吟じたのである。それは谷間にこだました。

「すすめ　すすめ　ものども
　じゃまな　てきを　けちらせ
めざせ　てきの　しろへ
　オゴレス　たおすのだ」

幾度も幾度もスレンは歌った。

果たして、すべての兵は落ち着きを取り戻していった。

王は、兵士達を一時退け、自らと騎士団のみでそれらの幻と戦うことにしたのだった。

そう、その決意を抱いた時に東ナデフソルダでの勝敗は決まっていた。強い信念を抱く者達に妖かしなど効かない……。妖術の源である魔導士を一刀両断のもとに斬り捨てると、幻の戦士は消え去っていったのだった。

イヨーの策略は二重策だった！　東ナデフソルダ峡谷の幻影は時間稼ぎに過ぎなかったのである。

オゴレスは、サイダムの地の旧スレン城跡地を決戦の場と定めていた。そして、バサム帝国の誇る不敗将軍、重騎士ダリエイとその配下の重騎士軍団を置いていたのである。

その圧倒的なパワーには、はっきり言ってスレン軍に勝ち目はなく、重騎士と渡り合うにはスレン軍はまだまだ未成熟だった。

バサム自慢の重騎士に対して、彼等の得意とするところの剣技でのみ挑んだなら、あの懐かしいスレン城を少しも見ることはできないだろう……。

では、ここで相手に勝るものは？　そう、策略である。巧妙な作戦でしか、この戦いには勝利はないのだ。

結果、スレン王はこの地で勝利を収めた。この最大の難関をどうやって勝ち進んだか？　しかし、それは残念ながら資料がないのである……。現在でもボコスカウォーズ史上最大の謎として学者達の間で論議されている。

わかっていることは、バサムのダリエイ将軍はまさに重騎士らしく豪壮であったが、実直にしてやや単純なところがあったということ。彼等はまた、バサム騎士としての誇りを持っていて、きちんと名乗りを上げ、申し合わせをしな

ければ戦いを始めようとはしなかったということである。

そしてこんなことを記すこと自体が罪なのかも知れないが、戦いにおいて、たとえば……、たとえば……「相手の背後から攻撃する」ことは、肉弾戦華々しいこの時代にあって、これほど効果的な手はないということである……。

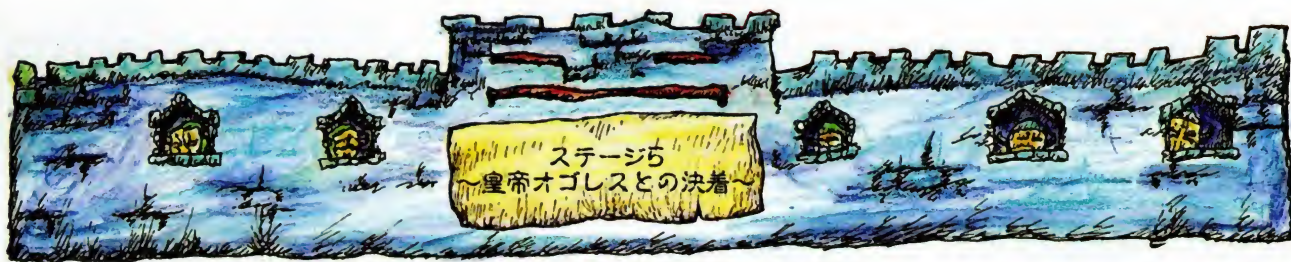
筆者自身は、ここでスレンが何か卑怯な手を使ったとは考えたくない。別の手段を講じたと信じている…。おそらく壁などをうまく使いすり抜けたのであろう……。

だが、相手の揺動に使ったその壁こそ、見るも無惨なそのかけらこそ、スレンの城壁だったと知ったスレン軍の心境はどうだったか……。スレン城はそこまでに荒らされ破壊されていたのであった。

「ここがスレンの城だったのか……」

戦いの最中にそれに気付き、戦いを終え、勝利してなお、スレン軍一同は、がくりと膝を落とし、首をうなだれたに違いない。

ボコスカウォーズの歌

ゲーム中に流れる音楽はなかなかいい効果音となるときもありますが、負けているときなどイライラしてしまうこともあるでしょう。そんな場合はボコスカウォーズの歌を歌いながらゲームをすると、もしかしたら調子よく勝つことができるかもしれませんね。（マニュアルより）

復讐に燃え、前線を放棄し、再びスレンを追撃した兵士長フィであった。が、長い戦闘を経験し、実力を本当のものとしたスレン達の敵ではなかった。フィ軍をジダム湖畔の戦いにて一蹴し、いよいよスレン達はアイアイダ山麓に位置するバサム城に乗り込む事になる。

バサム城。それは迷宮であった。オゴレスはその最深部「ヒサシカラズの間」に鎮座していた。彼と会いまみえるには重騎士軍の残党達とオゴレスの直属の親衛隊、そしてあの魔導士イヨーをも倒さなければならない。そう、それはスレンにとってこれまでの行軍の、全ての戦略の集大成となったはずである。後にも述べるが、実はこの決着もまた歴史の謎となっている。筆者がもう考えるのが面倒だからではない。謎だから謎なのだ。謎だと言ったら謎なのである。

……子どものようなもの言いは止めにして「オゴレス親衛隊」特色について知り得ている事を記述しておこう。「親衛隊長グディノウ」を長としている親衛隊は魔導士イヨーの魔法によって「ほとんど不死身化」している。強さで言えばスレンの精鋭の重騎士ほどの強さがあるのだ！魔導士イヨーは「強いものに強くあれ弱いものに弱くあれ」という魔法をかけているらしい。と、すればその攻略は？……読者諸君におまかせすることにしよう。

謎に包まれた結末～果たしてどちらが勝ったのか？～

さて、長々と書いてしまいました。いかがでしたか？　ボコスカウォーズ、これらを本当の史実のように感じて下さいましたか？

そうです。MSXを、MSXのゲームを何故今プレイするのか？様々に個々で好きなように想像して感じる事。可能性。その想像の源にパソコンが位置していた素晴らしい時代があったという事。かつて私達はMSXという提唱に希望の星を抱き、きっとボコスカウォーズに起つスレンの如き眼差でディスプレイを見つめていたのでした。今回それを思い出す作業をしてみたわけです。

今、貴方のパソコンは貴方の何ですか？

バサムとスレン。その勝敗はどうか貴方が決めて下さい。いつか貴方のストーリーを聞かせて下さい。多分その頃には夜逃げをしている私の耳に届くように。ではでは、また会う日まで。

イタチョコシステム　ラショウ

アイアイダの遺跡、参拝者と産廃車
アイアイダのオゴレス居城跡と思われる遺跡群。今は観光地としてイタチョコグッズ店ができたり潰れたりしている。

FROM DUNGEONS
HIT BUTTON OR SPACE KEY
UP TO 4 PLAYERS
AND
3 SPEED LEVEL
COPYRIGHT 1983 ASCII

ライズアウト

全面徹底攻略！ パスワードも全20面まですべて公開だ！

ここの攻略は、MSXマガジンに掲載されていたライズアウトの攻略記事を基にしている。

赤忍に気を付けろ！

ライズアウトは、キミの操作する白忍者を、敵である赤忍者に捕まったり、池に落ちないように誘導しながら、画面上へと続く"はしご"まで到達させるのが目的のアクションゲームだ。最初は、ドットで表現された、あまりに小さいキャラにびっくりかもしれないが、小さいなりに一生懸命走っている姿が妙に愛おしくなってくるから不思議。さて、そんなライズアウトだが、各面をクリアするのはなかなか大変だ。ちゃんと戦略を練らないと敵のはさみうちにあったり、どうやっても先に進めないってことになってしまうのだ。そこで、ここでは各面ごとに、ヒントを記しておこう。もちろん、これを読んでも頭脳と指先がついてこなければ始まらないぞ。そこは何度もチャレンジしてコツをつかむのだ！

LEVEL 1 ステージ名：なし

この面はさほど難しくない。鍵は○印の箱にあるので、それをまず取って脱出法を考えよう。

1面では敵の赤忍者の動きについて完璧に理解しよう。まず赤忍者の移動速度だが、キミが操作する白忍者に比べると約半分だから、そうたやすく捕まることはない。しかも、追跡ルールも簡単で、こちらと同じような動きしかしてこない。とはいえ、甘くみるのは厳禁。敵はひたすら追いかけくるし、死んでもすぐに最上部から復活してくるからだ。

LEVEL 2 ステージ名：SNAKE

ここでは敵の足の速さを知ろう。Ⓐ、Ⓑ地点では、特にそれに気をつかう必要があるぞ。

1面を無事クリアしたならば今度は2面。ここには扉がないので、ただ走りながら敵の忍者をやりすごせばいい。つまり、1面で覚えた赤忍者の特性を十分に利用すれば、おのずと道は開けてくる。ただし、相手はすぐに死んで最上部に復活してくるから、計2回もやりすごす必要がある。どこでやりすごしたらいいのかゆっくり考えよう。

【画面の見方】画面内の○で囲まれた部分は、クリアするための鍵が入っている。

レベル3で時間差ワザを磨こう

ライズアウトで、最初に登場する難解なパズル面がLEVEL3である。ここの右下にある箱は、どう考えても取れないのだ。ここは取らなくてもステージクリアはできるのだが、やはり取らなければゲーマーとしての名折れというものだ。実はここ、岩を崩して、それが復活することを利用した時間差ワザを使わないと取れないのだ。そこでライズアウトのおもしろさをいっそう深く理解していただくためにも、ここの攻略法を紹介しておこう。

ライズアウト

LEVEL 3 ステージ名：BEGIN

3面は、迷路がとても複雑になっている。しかも、ここでは鍵を見つけないと扉を開いてゴールすることができないのだ。鍵は○印のついた箱に入っているので、ほかの箱はともかく、これだけ取っておこう。そして、問題となるのは左上の階段での赤忍者の撒き方。うっかりすると追い詰められてしまうことしばしばだ。何度でもチャレンジしてほしい。

Ⓐ地点では赤忍者の誘導のしかたを身につけたい。Ⓑ地点で岩を溶かす。後は簡単なはず。

LEVEL 4 ステージ名：GRASP

ここは"渡り板"と"はしご"だけで構成された面。これらの特性を頭に入れないと、下に広がる大池にドボンとなってしまう。階段を上がることはそれほど難しいことではないが、降りるときや飛び降りるときには注意すること。ここには扉があるが、鍵を取って扉を開くという危険性を冒さなければ、ゴールにたどり着くことは難しくないのだ。

Ⓐ地点で敵をかわす。もちろん誘導しなければダメ。復活した赤忍者もⒷ地点でもかわす必要がある。

LEVEL 5 ステージ名：LADDER

この面には敵忍者が2人。おまけに面の構成は"はしご"だけ。でも、あまり深刻に考える必要はない。とにかく足は白忍者の方が圧倒的に速いのだ。敵がどこにいるのかを頭に入れながら、次にどの"はしご"を渡ればいいか考えよう。池に落ちないように気を付けてさえいれば簡単にクリアできるぞ。

ここではとにかく走り回ること。なにせ敵より足は速いんだから捕まりにくいワケだ。

LEVEL 6 ステージ名：TWIN TOWERS

さて、この面ぐらいになると、どこを通ればゴールにたどりつけるか、ちょっと頭を使わなければわからない。ポイントは敵の足の鈍さと、岩を溶かす銃の使い方。階段をうまいこと利用すれば赤忍者を撒くことなんて簡単なもの。この面ではすべての箱をいただいちゃうことができるのだ。

ポイントは"はしご"を使っての敵のかわしかた。Ⓐ地点もⒷ地点も要領は同じだ。敵は足が遅いのだ。

LEVEL 7 ステージ名：ASCII

この面には箱がいっぱいある。できれば全部取ってクリアしたいところだ。しかし、最初に問題となるのは下に並んでいる箱。白忍者の方が足は速いとはいえ、赤忍者2人に追いかけられたのではたまったものではない。ギリギリのタイミングでかわしながら、これまでの経験を活かしてガンバろう。

ⒶとⒷの箱を取りたいときには、赤忍者を反対側の階段上まで引き寄せてから一気にダッシュしよう。

LEVEL 8 ステージ名：LOOPS

この面は、大きく左右のエリアに分かれている。左側から右側に行く道は1カ所だけ。いったいどこを通ればいいのか考えよう。左側にいる限り、赤忍者は同じところを移動しているだけなので慌てる必要はない。それと、この面の箱の中には、時間差ワザを使わないと取れないものがたくさんあるのだ。

ここで気を付けなければならないのはⒶ地点の渡り板を降りるタイミング。チャンスは一瞬だぞ。

右下の箱を取るだけならばここから飛び下りればよいのだが、問題はその後。一歩左に動くと池にまっさかさま。右の岩を崩しても同様に池に飛び込むしかないのだ。悔しい。

さあ攻略本番だ。まず、赤忍者をこの位置に誘導しておこう。これ以外の場所だと赤忍者が白忍者の邪魔をして、ゆっくり攻略に集中することができないからだ。

攻略の第一段階は、白忍者の左の岩を、サブマシンガンで崩すこと。その先の岩まで壊さないように回数を数えて連打だ。また、その後は岩が復活するまでしばらく待とう。

岩が完全に復活する直前に、箱の下の岩を崩す。これで箱を取ったあとの逃げ道を確保するのだ。続けて箱の右上の岩を崩し、箱を取るための入り口を確保する。さあ急げ。

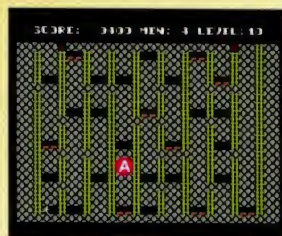
第一段階で崩した岩が完全に復活したら、あとは簡単。その上を通って確保した入り口から降り、箱を取って、やはり確保した逃げ道をさっと通ってはしごに戻る。できた？

LEVEL 9 ステージ名：ROOMS

この面の一番難しいところは、鍵入りの箱の取り方。在処はわかっていても、ちょっと頭を使わなければそこまでたどりつけない。左側の赤忍者を下の密室に閉じこめた後、岩を溶かしながら鍵のある箱までたどりつくのがコツ。右下の箱を取っていると、残る敵に捕まりやすくなるぞ！

Ⓐ地点に敵を誘い込み、その後でⒷ地点を通って鍵を取りに行く。敵はついてこないから安心だ。

LEVEL10 ステージ名：AMIDA

岩と"はしご"だけで構成された面だから、どこかに飛び降りたり、岩を溶かして敵から身を守ることはできない。だから利用できるのは敵の動き方のパターンだけ。この面にたどりつくまでにそれは十分に身に付いているはずだから簡単にクリアできちゃうよね。敵は2人とも左下の回廊に閉じこめちゃおう。

Ⓐ地点の回廊を利用して2人の敵を撒く。敵は1人でも2人でも同じようにして撒いてしまおう。

LEVEL11 ステージ名：FIFTH TOWER

立派な五重塔である。ここでは箱のある隠し部屋の中に敵がいる。だからこの敵を部屋から出さなければならないのだ。左側の"はしご"から隠し部屋の中の"はしご"のある最上階の部屋の岩壁に穴をあける。ここから赤忍者を外に出すんだけど、放っておくと自殺をしちゃう。彼らが降り始めたらすぐ隠し部屋に潜入しなくてはならないぞ。

Ⓐ地点の空間に敵を誘い出す。これでもう安心して鍵を取れるというわけ。あとは簡単だ。

LEVEL12 ステージ名：IN THE LAKE

レベル10と、かなり似通った面だけれど、あの面と同様に敵2人を左の回廊を利用して閉じ込めることができる。10面よりもむしろ簡単なぐらいに感じるかもしれないが、それはキミがこのゲームのコツを掴んできたためだ。この調子でガンバれば20面すべてをクリアするのも夢じゃない。ガンバレ。

10面と同様、Ⓐ地点の回廊を利用してⒷ地点に敵を誘い入れる。2人とも同じ手にのってくるのだ！

LEVEL13 ステージ名：GRAVEYARD

さて、13面。この面では鍵を取るのは簡単なのだが、その後の脱出方法が問題となる。まずは十字架の上を開け、そこから飛び降りながら鍵を取る。その後で、敵を一度十字架の中に閉じこめて出られないようにしたところで、初めてこちらも十字架の左側から中に入り、ゆっくり脱出するというわけ。ちょっと難しいぞ。

鍵を手に入れた後、敵を十字架の反対面に入れる。その間にこちらは中に入って脱出する！

LEVEL14 ステージ名：PYRAMID-2

ここも鍵を取ることよりも脱出することのほうが難しい面。しかし、敵を2人とも左側に誘導する方法を考えつけば、意外と簡単にクリアできる。赤忍者はこちらの動き方によって思いどおりにコントロールできるってことを頭に入れておきたい。そうすればなんてことはない面だ。

島の左側に敵を誘導してしまえば後は簡単。赤忍者はすぐ死んでしまうので行動は素早くね。

キーワードで好きな面にジャンプ

画面上の箱を全部拾ってからクリアすると、その後でキーワードが現れる。このキーワードを使うと、なんと好きな面に直行できるのだ。ここでは特別にキーワードを全部公開しておこう。どのように利用するかというと、具体的にはゲーム中に[CTRL]＋[P]キーを押す。すると、画面の上部に「KEYWORD?」と表示される。ここで4文字のキーワードを入力すればオッケー。難しい面に何度もチャレンジしてくれ。

レベル	パスワード	レベル	パスワード
1	——	11	5TOW
2	SNAK	12	LAKE
3	BIGI	13	GOLG
4	GRAS	14	PYR2
5	LADR	15	PYR3
6	2TOW	16	OOBA
7	ASCI	17	PYR1
8	LOOP	18	PYR4
9	ROOM	19	TRAP
10	AMID	20	TANK

ゲーム中に[CTRL]＋[P]キーを押すと画面上に「KEYWORD?」と表示されるのでキーワードを入力しよう。

超裏ワザ！分身の術!?

何度やっても敵に捕まってしまって、なかなかクリアできず、しかもGAME OVERでタイトルに戻る時間も待ちきれないというアナタに朗報！

この裏技を使えば、プレイヤーキャラである白忍者がどんどん増えてしまうのだ。方法は簡単。[CTRL]キーと[SHIFT]キーを押しながら、[2]キーを押すというもの。これで安心してプレイできるゾ。

ライズアウト

LEVEL15 ステージ名：PYRAMID-3

Ⓐ地点から岩を溶かす。岩は2発目までは復活しないので、その時間を利用しよう。

そろそろ難しくなってきたぞ。でも、鍵の在処がわかっていれば、全部の箱を盗み出さなくてもOK。狙いを決めればクリアは簡単。この面では敵を左側におびき寄せておいて、今度は上の方の岩をすべて溶かしてしまう。そこに飛び降りれば鍵が手に入るってわけ。一度右に落ちると左に戻れないから注意しよう。

LEVEL16 ステージ名：PERVERSE

Ⓐのはしごを使い、岩を溶かす。溶かした岩が復活する時間差を使えば鍵はすぐに手に入るのだ。

ここでは敵は1人と考えよう（つまり、右下の隠し部屋の箱を盗み出す必要はないのである）。そうすれば、後は簡単。岩が溶け、それが復活するまでの時間を利用して鍵を手に入れ、その後は脱出方法を考えるだけだ。一見複雑なようだが、ちょっと考えれば答えは見つかる。落ち着いて操作しよう。

LEVEL17 ステージ名：PYRAMID-1

Ⓐ地点に敵を閉じ込めてしまえば、後は作業がしやすい。岩を時間差で溶かし、中に入ろう！

ここが一番難しい。まず敵は左側の岩のクボミに閉じこめてしまおう。これは慣れれば誰にでもできる。その後で岩を溶かして中に潜入するのだが、そのためには「岩は2発目までは復活しない」という特性を利用する。この特性をキッチリと利用しなければ鍵は絶対に手に入らないぞ。ガンバレ!!

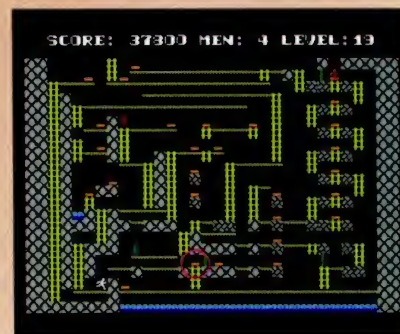
LEVEL18 ステージ名：PYRAMID-4

Ⓐ地点の扉をあらかじめ開けておきさえすれば、あとは足の速さで敵を撒くことができる。

もう、ここまでくればこのゲームをマスターしたのも同然だ。落ち着いてプレイしよう。さて、ここでは敵の忍者との足の速さの差を利用しよう。あとは池の真下の扉をあらかじめ開けておき、もう一度スタート地点に戻る。二度目は開けておいた扉の上から飛び降りれば鍵は簡単に手に入っちゃうぞ。

LEVEL19 ステージ名：TRAPS

落ち着いて迷路を解こう。鍵は一度右側まで行って取ってこなければならないのだ。

ゴールは近いぞ。でも、ますます迷路は複雑になってきた。こうなったらすぐにゲームを始めずに、しっかり作戦を立てるべきだね。よく考えれば答えは出てくるはずだ。この面ではゴールのすぐ近くまで行ってから鍵を手に入れ、それからもう一度スタート地点に戻る必要がある。焦らずに……。

LEVEL20 ステージ名：WATER TANK

Ⓐ地点を落下しながら撃ち溶かし、出てきた敵はⒷの"はしご"を利用して隠し部屋に閉じ込めちゃえ。

さあ、最後の面だ。ポイントは敵の1人を外に出し、さらに反対側の密室に閉じこめてしまうというところ。そのためには、落ちながら島との境の岩を打ち落とさなければならない。でも岩が溶けなければ敵は外に出ない。安心してチャレンジしよう。残った敵は足の速さが違うから怖くはない。早く"はしご"を上れ！

テセウス

18年後の今、初めて明かされるテセウスのスムーススクロール技術！ もちろん後半2ページではしっかり攻略もするぞ！

TEXT／三橋正邦
テセウス、イリーガスの他、AXシリーズ、シルフィードのオープニングの作曲などを手がけていた人物。現在はゲーム学校で講師などもしている。

MSXでスムーススクロールを

MSXは、その登場時、スプライトを手軽に扱えることがセールスポイントの1つになっていた。

確かに、スプライトは背景から独立し、ドット単位で簡単に動きをコントロールできた。PC-8801やPC-6001などとは違い、手の込んだグラフィック描画ルーチンを書かなくても、手軽に固定画面でのシューティングゲームを作ることができたのだ。しかし、画面全体をスクロールさせるゲームでは、背景スクリーンの複雑な構造により、8×8ドットのキャラクタ単位で行うものがほとんどであり、それらのゲームでは背景の動きがなんともぎこちないものにならざるを得なかった。

ところが、ゲームセンターに行けば、多くのアーケードマシンが滑らか全画面スクロールを実現している。その滑らかスクロールを何とかMSXで行うことはできないものか。それも、上下のみや、左右のみではなく、全方向のスムーススクロールをなんとしてもMSXの画面上で実現してみたい。テセウスの開発はそうした動機で始められた。

テセウスのキャラクタコード

MSXの背景スクリーンは、8×8ドットのキャラクタ単位で管理されている。縦あるいは横の、どちらかのスムーススクロールを実現するだけでよければ、スクロール方向に1ドットずつずらした8種類のキャラクタパターンを持てばいい。しかし、縦横両方のスクロールを実現するには、どんなに頑張ってもキャラクタパターンが不足してしまう。このためテセウスでは背景のスムーススクロールはキャラクタパターンの再定義を繰り返すことで実現している。

図1は、テセウスの背景スクリーンのキャラクタコード表である。0x00から0x1fまでがタイトルロゴ、0x30から0x5fまでがメッセージやスコア表示用の文字、0x60から0x6fまでがゲージなどに用いられるグラフィックパターン、そして0x80から0x98までがマップ描画用に用いられるエリアである。

図2は、マップ描画に用いるキャラクタの一覧で左側が通常の壁や床に用いられるキャラクタ、右側は放射性物質貯留池、通称「血の池」の描画に用いられるキャラクタである。

なお、プレイヤーキャラ、敵キャラ、マップ上のアイテムはすべてスプライトで描画している。

4×4のブロック単位によるマップ管理

テセウスでは、マップデータのメモリ節約のため、図3に赤のラインで示した4×4キャラクタのブロックでマップを管理している

一面の大きさは最大で32×16ブロック（=256×128キャラクタ=1024×512ドット）だが、画面上に表示できるサイズは8×5ブロックとなる。

赤のラインで区切られたブロックを切り出した例が図4で、図3の画面上で鍵が表示されている右隣のブロックである。この例では右と上に壁が存在するが、実は壁の有無によるキャラクタパターンのバリエーションは多くはない。1つのブロックに出現するキャラクタのパターンは図5に示すAからHの8パターンで、中央の4ブロックは常に空白になる。

キャラクタパターンは、1ブロック1バイトで管理するマップデータを基に作成される。ブロック周囲の壁の有無とデータの具体的な対応関係は図6のとおりだ。

図4のように右と上に壁がある場合は、マップデータはbit7、6、5、0、2がオン、bit1、4、3がオフとなるので、マップデータは0xE6になる。

床があるだけで上と左右が抜けているブロックではbit2、3、4がオンになるだけなので、データは0x1Cとなる（ROM上では、このマップデータはさらに圧縮されている）。

VRAMに転送するデータは、このマップデータを元に4×4のブロック単位で作成し、毎フレーム描画されることになる。

各キャラクタの持つパターン

次に、キャラクタジェネレータに転送するデータを作成する。このデータも個々のパターンに着目して考えれば、さほど多くのパターンは存在しない。

たとえば、左上隅の位置（図5の

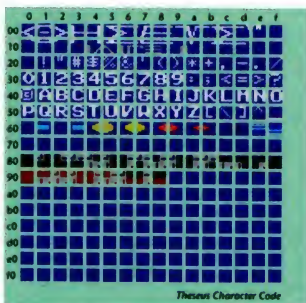



図1　背景スクリーンのキャラクタコード表

図2　マップ描画に使うキャラクター覧

注：0x00は16進数の00を意味する

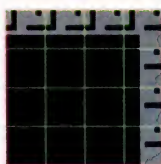



図4　赤のラインで区切られたブロックを切り出した例

図5　1つのブロックに出現するキャラクタのパターン

図3　スクリーンショットに、背景スクリーンのキャラクタパターンの境界線を加えたもの。赤のラインで区切った4×4ブロックでマップを管理している。

図6　壁の有無とデータの対応

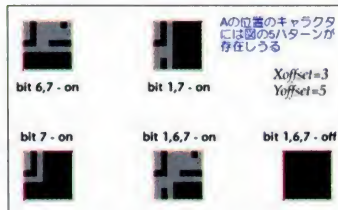
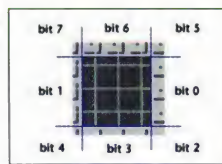



図7　図5のAの位置に使うキャラクタパターン

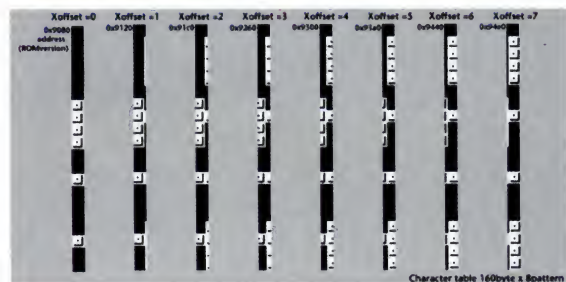



図8　壁に使用するグラフィックパターンのデータ

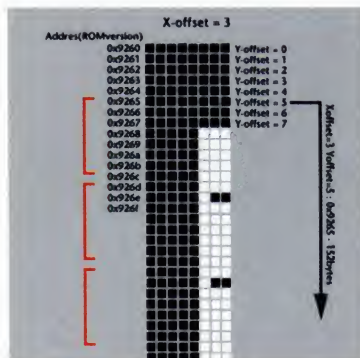



図9　図8の左から4番目のデータの拡大図。スクロールした分だけY座標分読み飛ばしてデータ転送する。

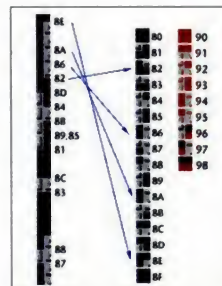



図10　グラフィックデータ(左)と、転送後にRAMに格納されたデータ(右)の対応。

Aのキャラクタ)に着目すると、マップデータのパターンによって、図7のように5種類のキャラクタが存在することになる。B、D、E、Gの位置ではさらに出現パターンは少なく、2種類のキャラクタしか存在しない。

これらのパターンは、画面のスクロール量によって表示されるパターンが変わってくる。ここでは最初に示した図3と同様に、スクロール量を8で割った余りである3がX方向を、Y方向が5の場合を示している。

キャラクタジェネレータへの転送データ

このキャラクタパターンを効率的に再定義するために、あらかじめ壁のグラフィックパターン用に図8の160バイト×8パターンのグラフィックデータを作成後、このデータを部分的にキャラクタジェネレータに転送している。

まず、X方向のスクロール量の端数が3の場合、図8のデータの左から4番目のデータを用いる。さらに、Y方向のスクロール量の端数分だけデータを読み飛ばす。具体的には図9のようにデータをずらして読みながらキャラクタジェネレータに転送する。

転送するデータは、図1のキャラクタコード表にもあるとおり、通常部分で16キャラクタ、血の池部分で9キャラクタだが、常時空白の部分は転送する必要がないため、実際の転送量は22キャラクタ、176バイトとなる。

とはいえ、この転送はリニアに一発転送……という状況ではなく、実際のプログラムでは図10に示すような複雑な関係になっている。これは、最初に作ったパターンとマップデータからVRAMにデータを転送するプログラムの間に不整合が生じたためで、データ構造を全面的に見直すよりは、ブロック転送のときにHLレジスタに加減算を行って調整する方が、場当たり的ではあるがバグを確実につぶせると判断したからだ。しかし、今からソースを見直してみると「もう少し何とかすればよかったな」と思える大きな反省点となっている。

アイテムや敵の描画

背景描画後、アイテムや敵、テセウス、お姫さまなどを描画する。

これらはすべてスプライトで描かれるので、技術的に困難な部分はない。スコア表示などのフレームにかかる部分を消すために上下のクリッピングが必要となるが、これは優先度の高い透明スプライトを置いて、はみ出し部分が表示されないようにしている。

そして15年……

当時はドット単位のスクロールを実現するだけで、こうした面倒なテクニックが必要だった。しかし、その後発売されるコンシューマ機には背景スクロール機能が搭載され、画面の開始位置を指定するだけで自由自在に背景画像を動かすことが可能となった。

テセウス発売からおよそ15年後の1998年、筆者はセガサターン用のBASICに同梱するサンプルアプリケーションとして、サターンBASIC用にリメイクしたテセウスを作成する機会に恵まれた。MSX版では、コーディング開始からスクロールプログラムが完成するまで、およそ2ヶ月かかった。しかし、サターン版ではサターンBASICの文法、画面構成などに関する知識がまったくない状態から始めて、スクロールルーチンが完成するまで2日しかかからなかった。考えてみると、MSX版テセウスは全容量が16Kバイト。現在ではフルカラー64×64ドットのアイコンが消費する容量である。当時の資料を見ながら、改めて時代の移り変わりを実感してしまった。

Close-up Game

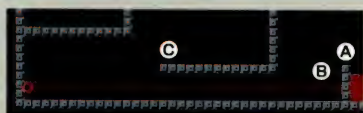
ゲーム攻略の基本中の基本テクを徹底図解！

さて、ここからいよいよテセウスの攻略方法を伝授しよう。

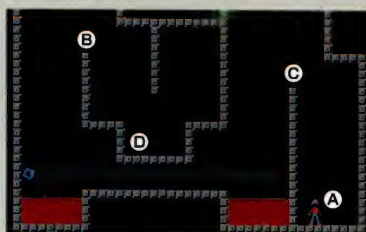
テセウスの目的は「指輪と鍵を取ってお姫さまを助けに行く」という単純なものだ。1、2面あたりではドアの対応も単純で、簡単にクリアすることができるはずだ

しかし、後半の面になるとさまざまなトラップが用意され、さらに敵を避けるタイミングもシビアになるため、攻略に苦労することだろう。

ここではまず、「一方通行」「閉じるべきドア」「飛び込みのタイミング」など、テセウスの代表的な攻略ポイントを図解する。

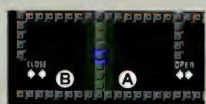
追いかけっこ（1面）

高速で動く敵の回避パターン。Ⓐから©へ移動する場合、敵がⒷの位置に来た瞬間に敵を追いかけるように飛び出す。止まらないで走れば、敵が左端で跳ね返ってくる前に©の位置に飛び乗ることができる。この面では、©からⒶの移動も要求される。

離れたポイントからの踏切り（9面）

ここでは©からⒷに移動する際、©の位置に立つと下を移動する敵が見えないため、Ⓐの位置でタイミングを計って飛び出す必要がある。逆の移動ではⒹから飛び出す。

高速移動する敵とパウダー（12面）

12面の中央下には、このように超高速で移動する敵に守られたOPENスイッチがある。この状態ではダメージ無しでドアを開けることはできない。このため、右図にあるパウダーを使って敵の動きを止める必要がある。

一方通行（8面／11面）

こちらの場面では、ⒶからⒷへの移動はできるが、ⒷからⒶに行こうとすると、どうやっても池に落ちてしまう。

上の場面では、Ⓐ側からⒷ側へ通り抜けることがきるが、逆はドアが閉じてしまうため移動できない。

まずは基本パターン（1面）

スタート地点の上にある「鍵」と、面の左下隅にある「指輪」を取り、「OPEN」でドアを開いてお姫さまの所に行けばクリア。六角星は999を上限にLIFEをプラス。旗は点数になるだけなので、クリアだけが目的の場合、特に取る必要はない。

飛び石と隠し扉（11面）

後半の面では、このように1キャラクタ幅の足場を飛ぶことも要求される。滑らずに確実に止まれるように練習しよう。右隅には隠し扉も見える。

閉じるべきドア（11面）

この例では、ⒶからⒷにダメージ無しで移動するためには、途中2カ所のドアが閉じている必要がある。

狭い通路・飛び込みと飛び出し（12面）

左の画面では、お姫さまにたどり着くために、テセウスの身長と同じ高さの通路に飛び込むのだが、これが結構難しい。[→]キーを押しっぱなしにしてタイミング良く[スペース]キーを離すのだが、失敗すると天井に頭をぶつけて落下してしまう。何度も試して練習するしかない。また、落下速度をゼロにできないため、落ちながら狭い通路には飛び込めない。しかし、狭い通路から落ちずに飛び出すことはできる。右の例では、ⒶからⒷへ移動することはできないが、Ⓑから©への移動は練習すれば可能だ。

テセウス

全ステージ攻略紹介!

ここでは各面の攻略法を解説する。スペースの関係でワンポイントアドバイス程度になってしまうが、各面のマップなどを参考に全面クリアを目指そう。

テセウスでは、4面、7面、10面クリア時にそれぞれ以下のメッセージが表示され、途中の面からスタートすることもできるぞ。

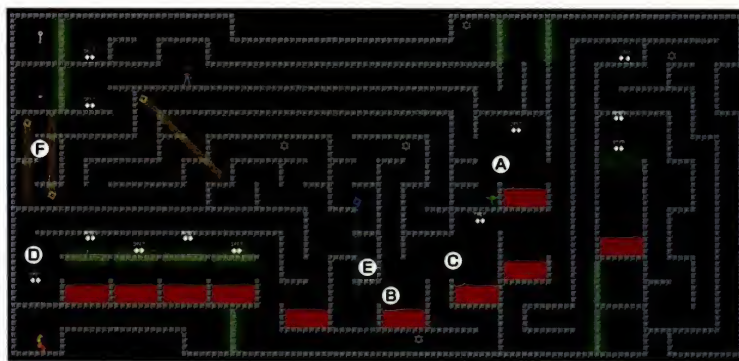
ENTER [7][5] FOR LEVEL 4
ENTER [0][1][7] FOR LEVEL 7
ENTER [5][3][1] FOR LEVEL 10

※実は、この数字は開発者の誕生日(AKIRA TAKEUCHI 7月5日、HIROMI OHBA 10月17日、MAO KOMINE 5月31日)から取られている。

面	攻略
1面	まずは練習。空中での方向転換やジャンプのタイミングなど、基本操作を覚えよう。
2面	左下では見えない位置で往復する敵を避ける必要がある。
3面	基本的な回避パターンを覚える。右上のドアは無理に開ける必要はない。
4面	パウダーは見えるだけ。いちばん下の敵は飛び越えてかわす。
5面	複雑なドアスイッチの対応。どのスイッチがどのドアに対応するのかを考える。狭い通路に飛び込む練習もここでやっておく。
6面	高速で動く敵との追いかけっこ。ギリギリのタイミングで飛び込め。
7面	敵の配置は4面と同じ。パウダーに触れると敵が止まる。
8面	隠し扉。どこがドアだかわからない。さらに、ドアは開ければ良いとは限らない。
9面	スタートポイントからの脱出にはシビアなタイミング取りが必要。しかも一度スタート地点に戻る必要がある。途中、時計で敵を止めてドアオープン。左下の指輪と鍵を取るのも難しいタイミングで敵を避けることが要求される。
10面	敵やアイテムの配置は5面と同じだが、ルートはより複雑。開けない方が良いドアもある。※詳細は図解攻略を参照。
11面	閉めるべきドアが3カ所、隠し扉が4カ所、一方通行が1カ所。※詳細は図解攻略を参照
12面	パウダーを使って敵を止める。狭い通路への飛び込みをマスターしていないとつらい。
13面	狭い通路への飛び込みのオンパレード。ルートも長い。時間切れにも気を付けろ。※詳細は図解攻略を参照

【10面】右上隅のスイッチで右中のドアを開けて下に向かう。指輪を取った後、下の通路を1周してから中央の部屋に向かう。左側に鍵の手前をふさいでいるドアスイッチがあるので、開けて鍵を取りに向かう。その後、下から2段目の通路に回って、中央の部屋の右側ドアを開け、マップ上端のドアスイッチに触れ、お姫さま救出に向かう。狭い通路からの飛び出し、飛び込みをマスターしていれば、旗の位置から鍵に飛び込めるので経路が短縮できる。

【11面】まず、右上のブロックに向かう。右側中央部分は、ドアを閉じながら進むとクリアしやすい。右上隅のドアスイッチはCLOSEすること。その下の2カ所のスイッチはお姫さま前の隠し扉スイッチなのでOPEN。一方通行のドアを通って中央に。パウダーで敵を止めて直上のCLOSEスイッチに触れるが、前もって右下隅のOPENで扉を開けておくと脱出しやすい。左上ブロックで鍵を取り、指輪前のドアを開け、右下隅に移動する。そこで指輪を取ってお姫さま救出に向かおう。

【13面】スタート地点から左に向かい、Fの通路を降りた後、最初の分岐を上に。Aを足場に隠し扉を開けておく。いったん戻ってEの通路を降り、Bから踏み切って右上に向かう。この踏み切りはかなり難易度が高い。Cの足場から上の扉を開き、最下段の通路からDをめざす。ぐるりと回って池に落ちないように踏み切り、スタート地点に戻る。右上のCLOSEで足場を確保し、鍵の前の扉を開く。鍵と指輪を取った後、F、E、Bを経由して左下のお姫さまに。

MSXPLAYerのBASIC版には、フロッピーディスク4枚分の仮想フロッピーディスクが装備されている。その1枚目(ディスクA)には、かつてアスキーから刊行されたポケットバンクシリーズやMSX関連書籍に掲載されたサンプルプログラムが収録されているのだ。市販のゲームプログラムのような派手さはないけど、けっこういい味出してるので、試してみてチョ！

MSXプログラム集

手作り感覚が味わい深いぞ

TEXT：高橋秀樹

[プログラムの種類]
 ゲーム
 環境ソフト
 開発ツール
 実用アプリ
 ユーティリティ

MSXPLAYer BASIC版

はじめに、MSXPLAYerのBASIC版の種類について触れておこう。BASIC版には、さらに、ジョイパッド版とマウス版の2種類があるのだ。スタートメニューから開くメニューの[MSX MAGAZINE 2003]→[スキン名]の下に、[BASIC(ジョイパッド版)]と[BASIC(マウス版)]という2つのプログラムが登録されているはず。

2つの使い分けは簡単だ。マウスを利用しなければならないプログラムを起動するには、マウス版を起動し、それ以外のプログラムを起動する場合には、ジョイパッド版を使えばいいのだ。

プログラムの実行方法

さっそく、BASICで書かれたプログラムを実行する方法を紹介しよう。

プログラムをディスクから読み出して、そのまま実行するにはRUNコマンドを使用する。RUNには、引数にファイル名を指定することができるが、このときファイル名をダブルクオーテーションで囲む必要がある。

たとえば、「ACGAME1.BAS」というプログラムを実行したければ、次のように入力する。

```
run "acgame1.bas" [リターン]
```

最後に[リターン](Windowsの[Enter])キーを押すのを忘れずに。なお、大文字と小文字のどちらで入力してもかまわない。

プログラムの終了方法

実行中のBASICプログラムを停止させるには、[CTRL]＋[STOP](Windowsの[Ctrl]＋[PageUp])キーを押す。

ただし、これはプログラムを強制的に終了させる方法なのだ。ゲームやBGV(バックグラウンドビデオ)などのプログラムでは、プログラム中で使うデータを保存する必要が生じないものが多いので、この方法で終了してもかまわないが、実用プログラムや開発ツールなどは、この方法で終了してしまうと、作成したデータが保存されないことがあるので注意しよう。データを保存してから終了させなければいけないプログラムでは、ほとんどの場合、終了方法がプログラム側で用意されているはずだ。

では、さっそくいろいろなプログラムを実行して、じっくり味わってほしい！

BASIC版にはジョイパッド版とマウス版がある。どちらも、入力インターフェイス以外に差はなく、画面のようなBASIC環境が起動する。

海底大戦争

▶▶▶▶▶ ACGAME1.BAS

海底で極秘に繰りひろげられる、無言のゲームだ。キミは、某国のミサイル基地を攻撃すべく、公海を航行中だ。しかし、敵陣が近づくにつれ、海面下には潜水艦がウヨウヨしてきた。でも、大丈夫。こっちには、高性能のソナーがあるから、海面下がどんな状況なのか一目瞭然だ。敵国の潜水艦を破壊せよ！

［スペース］キーを押すとゲーム開始。画面には、ソナーからの信号を分析して作った海面下の状況が表示される。海上に浮かんでいるのが、キミの艦で、海中に現れる黄色い潜水艦が、敵艦だ。敵の攻撃を避けながら、爆雷を落として敵をやっつけよう。艦の移動は、左右の矢印キーで行う。［Z］キー、［X］キーを使うと、艦首と艦尾から、それぞれ爆雷を投下することができる。

起動時のマシン語読み込み待ちは∞モードにしておくと速いぞ！

敵潜水艦の魚雷を避けながら、爆雷を投下しよう。敵潜水艦には、高速タイプと低速タイプがあり、特に海底すれすれを行く高速艦の撃沈は難しい。

スクェアランナー

▶▶▶▶▶ ACGAME3.BAS

真四角な区画と、縦横に等間隔で作られた道路。この無機質な未来都市に、エイリアンが出没するようになり、外出禁止令が出る始末。しかし、何者かの運転する謎の車が現れ、突然、エイリアンをやっつけ始めたではないか。人々は、深夜のこの救世主をスクェアランナーと呼び、心の中で惜しみない喝采を贈るのだった。

［スペース］キーを押すとゲーム開始。キミは、深夜のヒーロー、スクェアランナーを矢印キーで運転する。ランナーは、道路のゴミを拾いながらひたすら突っ走る。ある程度ゴミが溜まったら、中央にあるガソリンスタンドで給油しよう。エネルギーがいっぱいになると、エイリアンが青くなってひるむので、そのスキに車でひいてやっつける。スクェアランナーは、勢いで走ることしか知らないので、調子に乗って走り過ぎると、すぐにエイリアンに食べられちゃうぞ。

できるだけ、エイリアンを四隅に引きつけておき、画面中央のガソリンスタンドで給油しよう。ガソリン給油後、一定時間はエイリアンがひるむけど、調子にのってると復活しちゃうぞ！

トラッドレース

▶▶▶▶▶ ACGAME4.BAS

懐かしいタイプのカーレースゲーム。特徴は、自分以外の車が、周辺の状況を把握しながら走っているということ。前の車が邪魔なら、その車を避けて追い越したり、避け切れない場合は減速する。結構、リアルなレースが味わえるはずだ。

キミの車は、最後列左側の青い車だ。車の操作はジョイスティクか、矢印キーで操作する。左右がハンドル、上下は速度だ。制限時間内に距離30000を走り抜き、ゴールすれば、次のレースにチャレンジできるのだ。得点は、走った距離と、ゴールしたときの残り時間で計算される。

コースに車が並んだら、トリガーボタンか［スペース］キーを押し、全車スタート。他車と接触すると路肩にはじき飛ばされ、停止してしまうので、路肩で十分加速してから、後続車に注意してコースに戻らなくてはいけない。芝生では、スピードが出ないので注意！

時間内にゴールできないとゲームオーバー。クラッシュ後にうまくコースに戻れるかどうかがポイントだ。第５レースからは、白い根性悪の車が出現するぞ！

MSXプログラム集

最終インベーダー

1978年ごろ、突如日本列島に侵略してきたインベーダー。当時、多くの勇者たちがインベーダーの侵略を阻止すべく日夜戦い続けた。

インベーダーゲームの“最終形”を目指して作られたこのゲーム。スピードといい、キャノン砲といい、最終の名にふさわしい仕上がりになっている。マシン語プログラムならではの動きをよく見てほしい。感動ものだぞ。

キャノン砲は5台。防御ブロックの陰を上手に使って、インベーダーの攻撃を避けながら、攻撃をかけるのだ。キャノン砲の移動は、矢印キーで、ミサイルの発射は[スペース]キーで行う。キャノン砲がインベーダーの放つミサイルに衝突するとやられてしまい、倒したインベーダーも復活してしまうので、注意が必要だ。

もちろん、画面上にときどき出没するUFOを狙えば、ボーナスがもらえるので、高得点につながる。

自機がやられると、せっかくそれまでに倒したインベーダも復活してしまい、態勢を整えてくる。防御ブロックで敵ミサイルの猛攻をうまくかわそう。

星間特攻隊

ACGAME7.BAS

太陽系から0.5光年離れた宇宙。そこへ地球攻撃に向かおうとするUFOの大軍団が、密かに集結していたことを、地球防衛軍が突き止めた。ところが敵は、回りに小惑星群を配置し、強力な防御体勢をとっていて、外部からの攻撃は容易ではなさそう。腕利きのパイロットが集められ、最後の切り札、星間特攻隊が結成された。それぞれ複雑な思いを胸に、地球を旅立つ！

特攻機は5機のみ。[スペース]キーでいったん発射したら、もう後戻りはできない。ひたすら左右の矢印キーで自機をコントロールして、隕石群の間を避ける。そして、敵のUFOに突っ込むのだ。このゲームはかなり難しくできているので、第一級のゲーマーでも苦戦を強いられることは間違いない。難しければ、MSXPLAYerの[SLOW]ボタンで速度を下げてみよう。特攻機がすべてなくなったら、[Y]キーで再プレイだ。

隕石を避けるのも難しいが、その後UFOにうまく当てなければならないのだ。最初のうちは問題ないが、UFOの数が減ってくると、かなり厳しいぞ。

くじけちゃいけないバッティング

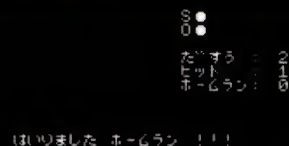
BATTING.BAS

最近、バッティングが今ひとつの赤ヘル君。なんと、バッティングセンターで特打ちを命じられる始末。情けなさと悔しさを胸に、ヒーローとして栄光をつかむまでひたすら特訓するのだ。

使い方は、バッティングマシンがボールを投げたら、タイミング良く[スペース]キーを押して、バットを振る。結果に応じて、5種類のメッセージが用意されている。ボールを見送るか、空振りならストライク。3アウトでゲーム終了だ。3本ヒットを打つと猛打賞が、全然打てないとかなりバカにされちゃうぞ。果たしてキミの成績はいかに？

プログラムは、SCREEN5というドット単位でパレットと組み合わせて最大512色の中から16色が選べるモードを使用している。美しいゲーム画面もちゃんと見てね。

単純だけど難しい。はまること確実だよ！

投球に合わせて[スペース]キーを押すだけなんだけど、これがなかなかジャストミートしないのだ。操作が単純なので、家族みんなで楽しめるぞ。

BASICサンプルゲーム ▶▶ BASGAME.BAS

MSX世紀9918年。地球は、宇宙の果てからやって来た、悪の帝王の配下であるUFO部隊によって攻撃を受けていた。地球連合軍はすでに壊滅し、もはや頼みの綱はMSX研究所で密かに開発された迎撃砲台Zだけだ。Zよ、地球の危機を救うのだ。

[スペース]キーを押すとゲームスタート。砲台は、ジョイスティックか左右の矢印キーでヨコに移動し、トリガーボタンか[スペース]キーを押すと、上に向かってビームが発射される。ビームはUFOに当たると"HIT!"メッセージが出る。また、UFOは左から右に移動しながら、下へ向かってときどきビーム爆弾を発射する。敵のビームが砲台に当たると派手なパレット切り替えのアクションとともに、ゲームオーバーとなる。シューティングゲームの基本中の基本なので、プログラミングの参考にしよう!

マシン語サンプルゲーム ▶▶ MACGAME.BAS

スプライトをマシン語でコントロールしているシューティングゲームだ。MSX-BASICのPUT SPRITE命令やSPRITE$関数は使用せずに、マシン語で直接VRAMにデータを書き込んだり、読み出したりする手法が使われている。このゲームでは、MSX2用のBIOSコールが使われているが、MSXPLAYerが、しっかり再現できていることがわかる。

[スペース]キーを押すとゲームスタート。砲台は、左右の矢印キーでヨコに移動し、[スペース]キーを押すと、上に向かってビームが発射される。ビームはUFOに当たると爆発する。また、UFOは左から右に移動しながら、下へ向かってときどきビーム爆弾を発射するぞ。

マシン語ならではの動きの速さを体験してみて欲しい。マシン語部分のアセンブラソースMACGAME.ASMも参考にどうぞ!

ゲーム「たぬきウォーズ」 ▶▶ GAME1.BAS

お腹の減ったたぬきくん。森をさまよっているうちに、大きな樹の根元にたどり着いた。自慢の出っ張ったお腹もすっかり引っ込んでしまい、もうヘロヘロ状態だ。なんとかエサにありつかなければならない。しばらく途方にくれていたたぬきくんは、最後の力をふり絞ろうと、とんでもないことを考えた……。

たぬきくんが、腹鼓(はらつづみ)の衝撃波を使って、画面上から落ちてくる虫(@)を採るゲームだ。操作は矢印キーと[スペース]キーのみ。矢印キーで左右に移動して、[スペース]キーで腹鼓を打つ。1回、衝撃波を使うとエネルギーが減り、たまに落ちてくるフルーツ(F)を取ると回復する。3分間の時間制限があり、その中で高得点を競う。

エネルギーを使い果たすと腹鼓が打てなくなるので、ご利用は計画的に!

タイニーシューティングゲーム ▶▶ GAME2.BAS

惑星MSXは、未知の宇宙人から侵略を受けた。人類は勇敢だったが、戦いは長期化した。決定的な敵への対処法が見つからないなか、偶然にも不時着した敵の宇宙船を手に入れることに成功した。はじめは未知の文明を目の前に全くお手上げの状態だったが、しだいにに成果が現れていった。そして人類は、ついに敵の宇宙船のコピーを大量に生産できるようになり、侵略を受けてからちょうど3年目の同じ日を控え、敵に大々的な攻撃を試みることにした。さあ、決戦は目の前だ。

プログラムをロードしてRUNするだけだ。ゲームが始まると、敵の宇宙船が空からやってくるので、左右の矢印キーで動きながら[スペース]キーでビームを発射して、打ち落とす。敵に当たっても自分が死ぬことはないが、そのあたりはご愛嬌。

3Dカーレース ▶▶ GAME3.BAS

カーレーサー、ジャック・スミスは苦悩していた。今季から新しく移ったF1チームでのテスト走行が全くうまくいかないのだ。自分のドライビングテクニックには自信がある。しかし、問題は来年から採用されるテスト・カーだ。早々にできあがったものの、どうも馴染めない。何度、走行と調整をしてもダメなのだ。このままでは、来季出場しても良い成績は残せない。なんとかしなければ……。

車のスプライトを複数の大きさで描いて、近くになるに従って切り替えることで、擬似3Dに見せている。縁石の色の変化は、パレットを切り替えているだけだ。

[スペース]キーがアクセル(ブレーキは無い!)。左右の矢印キーで、車を移動させる。ゲーム終了時に、クラッシュした回数と、走行距離が表示される。

こんなに短いプログラムで3Dに見えるなんて不・思・議・!

ヒント! 高速化で待ち時間を減らそう!

BASICプログラムを実行しても、すぐには何も画面に表示されてこなかったり、数字のカウントばかりでなかなか始まらないものもある。

これは、プログラムの初期化に時間がかかっているからだ。多くの場合、スプライトパターンの定義を読み込んでいたり、マシン語プログラムをメモリに読み込んでいたりするために起こるもの。

そこで、この待ち時間を減らすために、MSXPLAYerの[FAST]ボタンや[∞]ボタンを活用してみよう。これらのボタンは、画面書き換えなど、I/O(周辺機器との入出力)とのタイミングが取れないこともあり得るため、すべてのプログラムについてオススメというわけではないが、ここで紹介しているプログラムの初期化待ち時間を減らすのには効果絶大だ。

MSXプログラム集

ラージヒルジャンプ

▶▶▶▶▶ 90MJUMP.BAS

スキーのジャンプ競技の楽しさを少しでも味わおうというのが、このゲームだ。

実際の競技と同じように、風がやんだら、5秒以内に［スペース］キーを押してスタートする。選手は身体を抱え込んで台を滑っていくので、先端の傾斜の緩やかなところに差し掛かったら、もう一度［スペース］キーを押して、大空高くジャンプする。しかし、タイミングを間違えると、そのまま転倒したり、バランスを崩して早く着地してしまう。うまく踏み切れた後も、飛んでいる途中で、［スペース］キーをたたきながらできるだけ長く体勢を保たなければならない。

このプログラムは、選手が理想的な飛行曲線を描ける範囲を制限している。普通に飛んでいると、早めに範囲を外れてしまうので、［スペース］キーを押して、身体を押し上げるというわけだ。先に行くに従って、範囲は狭くなっている。

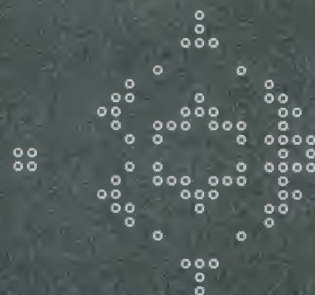
空中での姿勢は［スペース］キーを適度に連打して保つのだ。うまくいけば、120mくらいまでは飛べるから、感激すること間違いなしだ。頑張れ日の丸飛行隊！

ライフゲーム

▶▶▶▶▶ LIFE.BAS

生物の繁栄滅亡をシミュレートするゲーム。RUNさせると、四角いカーソルが表示されるので、これを矢印キーで動かし、［スペース］キーで“○”を配置する。これが、生物だ。いくつか○を配置したら、［リターン］キーを押して、シミュレート開始。生物の存亡は以下のように決められる。

1. 周囲8マスのいずれかに、2～3匹の仲間がいる場合、真ん中の生物は生き残る
2. 周囲8マスのいずれかに、4～8匹の仲間がいる場合、過密のため、真ん中の生物は死ぬ
3. 周囲8マスのいずれかに、1匹しか仲間がいない場合、孤立のため、真ん中の生物は死ぬ
4. 回りの8つのマスのいずれかに、4匹の仲間がいた場合で、真ん中に生物がいなかった場合、そのマスに生物が誕生する
5. 以上の規則は、すべてのマスについて同時に行われる

このゲーム、コンピュータサイエンスでは古典的なテーマなのだ。世代ごとに変わっていく様子を眺めて楽しもう。シミュレート中止は、［スペース］キーだよ。

パラシュートマン

▶▶▶▶▶ RAKKASAN.BAS

スパイ008は、某国の開発した秘密兵器のマイクロフィルムをひそかに盗み出したが、敵に察知されてしまった。果たして彼は無事に生還できるのだろうか。

ゲームがスタートすると、左から右に飛行機が飛んでいくので、適当なタイミングで、［スペース］キーを押して飛び降りる。パラシュートがすぐに開いていて、空の散歩としゃれ込みたいが、すぐに敵が現れる。敵は、キミの動きを執拗に追いかけてくるから、左右の矢印キーを押してパラシュートを操作し、接触を避けなければならない。間に合いそうもない場合でも、［スペース］キーを押せば、うまく避けることができるぞ。また、自分が敵のスピードよりも速い場合は、体当たりでやっつけることも可能だ。しばらくすると、地上が見えてくるから、“GOAL”と描かれた標的の上に着地する。判定がきびしいぞ！

パラシュート降下中は、左右の移動もゆっくりふんわり。急な方向変換はなかなかできないので注意しよう！ 着地点まで、一瞬たりとも気が抜けないぞ。

近代空中戦

▶▶ ACGAME6.BAS

最大4人まで対戦可能な空中戦ゲーム。キミたちは、特別な訓練を受けた戦闘機パイロットだ。今から基地上空で、空中戦の模擬訓練をする。勝ち残った者がエースパイロットだ！

RUNすると、最初にプレイヤーの数を聞いてくるので、2～4の数字を入力。さあ、ゲーム開始だ。画面に現れる各機のエンジン後方には、番号が振ってあり、プレイヤーは、各自の戦闘機に対応したキーで、戦闘機をコントロールする。PLAYER-1からPLAYER-4までの操縦キーの割り当ては、次の通り。

PLAYER-1：[1] [2] [3]
PLAYER-2：[9] [0] [-]
PLAYER-3：[Z] [X] [C]
PLAYER-4：[<] [>] [?]

左から「左旋回」、「ミサイル発射」、「右旋回」だ。

あっぷっぷ

▶▶ APUPU.BAS

「ニラメッコシマショ、アップップ！」

キミは、ドンナ顔ヲスルノカナ？ 最近、表情ノナイ人間ガフエテイルヨウダカラ、キカイノボクガ人間ラシサヲトリモドシテアゲマショウ。「スペースキーヲ、オシテクダサイ」

なんてことはない、ただのにらめっこゲームなのである。しかし、パソコンは絶対に声を出して笑ったりしないから、キミがどこまで耐えられるかが勝負だぜ。使い方はプログラムをロードして、RUNするだけだ。

プログラムは、顔のパーツをスプライトで定義したあと、[スペース]キーを押すたびに乱数でパーツを選んで表示している。単純な仕組みだけど、意外に笑えるのだ。

このゲームは、マシンの画面モードを800×600に切り替えて大きく表示したほうが効果的！

いざ、勝負！

パネルメイズ

▶▶ P_MAZE.BAS

2プレイヤーで競う、運が頼りの迷路ゲーム。8×8の64枚パネルの下には迷路が隠されている。パネルの下に道があれば進むことができる。相手より先に迷路を抜けたほうが勝ちだ。

RUNさせると、振り出しの位置に水色と緑色のコマが置かれる。水色からスタートだ。画面下部にめくるパネルの「よこ」と「たて」の座標を入力する表示が出るので、現在位置の上下左右のいずれかの座標を、カンマで区切って入力し、[リターン]キーを押す。するとパネルがめくられ、道があった場合、パネルがめくられ1歩進める。迷路は全部で5面。物足りないと思う人は、プログラムの2160行以降をいじってみて欲しい。データ形式は16進数ひとつが迷路のよこ1行分を表している（全部で8個）。9個目と10個目はゴールの座標だ。

複雑な迷路を作っておいて、みんなでワイワイやると楽しいぞ！

カメレオンハンター

▶▶ KAMEREON.BAS

8×8のフィールドのどこかに、カメレオンが隠れている。1ターンごとに適当にバラまかれているエサが消えるので、エサの消えたラインを手がかりに、どのマスにヤツがいるか当てよう。

RUNさせると、フィールドが表示されたあと、乱数で“十”の形をしたエサがまかれる。まず、キミの勘がさえているかどうかを確かめるため、ヨコ、タテの順番で、隠れているカメレオンの位置をカンマで区切って入力しよう（例、A,1）。もし、外れたら、こんどはカメレオンがえさを食べるから、それをヒントにして欲しい。もちろん、ひとりで遊べるけど、何回探したか手数も表示されるので、友達と競ってみるのもおもしろい。

なお、エサは毎回乱数で設定される。

入力が小文字だと認識されないので、MSXPLAYer側で[CAPS]キーをオンにしてプレイしよう。

楽しいカンけり

▶▶ KANKERI.BAS

カンけりのルールから説明しよう。仲間を4、5人集めよう。そして、缶を1つ用意して立てる。1人がじゃんけんでオニになり、目をつぶって数を数えているあいだ、残りがいっせいに隠れる。オニは、隠れた人を全部探し出して名前を呼び、急いでカンを踏む。探しているあいだは、当然オニが缶から離れるから、隠れている人はチャンスを見計らって缶を蹴るのだ。カンを蹴られたオニは、もう一度オニにならなきゃいけない。

RUNさせると、画面の真ん中に大きな缶とオニ（実はキミがオニなのだ）が表示される。カーソルキーで移動しながら、回りの木陰を探してみよう。どれか1本に人が隠れていて、うまく見つけて[スペース]キーを押し、缶を踏めばOK。誰もいない木を選ぶと、そのすきに、別の木に隠れている仲間の誰かが缶を蹴りにくるから、急いで戻って缶を踏もう。

豆占い

▶▶ MAME.BAS

占いといえば、手相や星占いなどが一般的だ。でも、日本の風土に根ざしたこんな占いもあるのだ。小正月や節分の夜、大豆を並べて、炉の中にくべて焼く。それぞれの焦げぐあいで1年の天気を占う。これは、日本の農耕に関連した占いで、豊作を願った昔の人々が編み出したものと言われている。

やり方は、プログラムをロードしてRUNするだけだ。[スペース]キーを押すと即、結果が表示されるぞ。白いものは晴れ、うす焦げなら曇り、黒焦げは雨、はじけたら晴れというわけだ。

プログラム内容は、スプライトを定義したあと、乱数ではじけ具合を決めて、結果を表示している。コンピューターの世界だから、今ひとつ信用できないっちゃあできないかも（笑）。

プログラムリスト中のメッセージを変えて別のことを占ってみてもいいかも!?

MSXプログラム集

チャップくんの玉入れゲーム

▶▶ TAMAIRE.BAS

メチャクチャ忙しい玉入れゲームだ。飛んでくるブロックを避けながら、空中のかごに向かって玉を投げ入れるのだ。キミに代わって、どんなときでも人を楽しませることを忘れないチャップくんが、愛用のステッキと帽子を手にしてチャレンジしてくれる。必死な姿は、きっとキミの胸を打つことだろう。感動の涙を流してもかまわないんだよ。

ゲームのルールは、次の通り。

RUNすると、かごとブロックが左右に移動し始める。[スペース]キーを押すと、チャップ君が自動的にカゴのほうへ玉を投げるから、方向は気にしないでかまわない。ブロックを避けるためのジャンプは、矢印キーの[↑]を使うんだけど、空中にいるときでも[←][→]キーで左右に移動することができる。

さあ、制限時間内に何個、玉を入れられるかな？

スモール・ワールド

▶▶ S_RPG.BAS

テキストベースのロールプレイング・ゲームだ。プログラムをRUNさせると、ゲーム画面がただちに表示される。主人公は“*”(アスタリスク) だ。画面上に散らばっているアルファベットで表されたモンスターを倒しつつ自分自身を成長させ、最後に敵のボスである「X、Y、Zの魔法使い3兄弟」をやっつけるのが目的だ。はじめに、主人公の個性を決めるために与えられた12ポイントをST(強さ)、AG(敏捷性)、DG(威厳)、IQ(明晰)に振り分ける。キャラクタができたら戦闘開始だ。戦いたいモンスターへ矢印キーを使って、主人公を移動させる。*とモンスターが重なったら、[スペース]キーを押すだけで戦いの結果が表示される。敵は、@が一番弱く、あとはアルファベットのA～Zの順に強くなるので、いきなり強い相手に戦いを挑むと負けるかもよ。

冒険の無事を祈る！

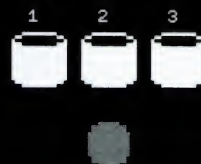
たまはどこ？

▶▶ TAMADOKO.BAS

素早く入れ替わる3つのコップの1つに入ったボールの場所を当てよう。プログラムをロードしてRUNするだけで、すぐに遊べる。たまをコップの中に入れる演出のあと、コップを動かす回数とスピードを聞いてくるので、それぞれ数字で回数を入力する。[スペース]キーで、いざ開始。激しくコップを入れ替えたあとに、答えを聞いてくるので、コップの上に表示された番号を入力しよう。はずれると「ざんねん、むねん……」と表示され、かな～り悔しい。

3つのコップは、それぞれ3枚のスクリーンに割り当てられている。左のコップから画面番号は0、1、2という具合だ(0が一番手前)。コップが入れ替わるたびに、変わってしまうコップの位置関係を初期状態に戻すので、移動ルーチンは同じものを流用している。

パーティ会場とかでやれば、盛り上がること間違いなしだ。

コップの入れ替えは50回まで選ぶことができるが、あまり多いと途中から追い切れなくなってしまう。それでも、結局3択だからいいんだけどネ。少ない回数なら、しっかり見て、当てられるぞ。

花火

▶▶▶▶▶ HANABI.BAS

今年は花火を見にいったかな？ 見にいけなかった？ そうかそうか。そんなキミのために、夏の風物詩の花火を、MSXでもできるだけリアルに表示できるようアルゴリズムに工夫を凝らしたエンターテインメントプログラムを紹介しよう。操作は、プログラムをロードさせてRUNさせるだけでOK。

ヒュ～、ドン！！ って音と一緒に美しい花火がスクリーンの夜空に咲くよん。お部屋を暗くしてみると雰囲気が出るかも。思わず「玉屋～」っと掛け声かけたくなるかもね？

とにかく、花火のリアルな動きをBASICだけで作るのは正直言ってかな～りキツイのは確か。そこで、ここでは円を描くアルゴリズムに「デジタル微分解析(DDA)」という手法を使って、描画スピードの高速化を計っている。音は、ノイズと高めの音を組み合わせて、それっぽく仕上げている。

次から次へと尺玉の連発で景気がいい花火大会だぞ。何か、おめでたいことがあったときに、ボリュームを上げてRUNしてみるのもグー！

も・よ・う

▶▶▶▶▶ MOYOU.BAS

パターンのデザインは、知識や経験がないフツーの人には、色や形をどう組み合わせればいいのか難しくて、もうアチョ〜って感じだ。でも、自分のホームページの背景なんかにオリジナルのパターンが敷き詰められれば、嬉しいよね。そこで、デザイナーでなくても、シームレスなパターンを作れるプログラムを紹介しよう。プログラムをRUNすると、一定時間ごとに何らかの模様を描いてくれる。気に入ったものができたら、[STOP]キーを押して、パターンやや色の構成を紙に記録しておけばいい。いちいち手描きするのが大変だって？　それならば、Windowsの機能でキャプチャしちゃえ！　それがMSXPLAYer流の活用術なのさ。

SCREEN1のキャラクタ定義機能を使って、1文字分だけランダムなキャラクタを定義している。同じく文字の色もランダムだ。

さまざまな色の組み合わせで、次々とパターンが表示されていく。ランダム表示だけに、突然すんばらしいデザインができることもあるかも？見逃しちゃダメよ。

アニマルどけい

▶▶▶▶▶ TOKEI.BAS

時計屋さんにいくと、カラクリ満載の楽しいアナログ時計にいっぱい出会える。中でもキャラクタをテーマにしたものは、時間をみるのはもちろん、眺めているだけでもチョー楽しい。

そこで、BASICで作ったオリジナルのアナログ時計を紹介しよう。テーマはアヒルの親子。文字盤の中心には、おかあさんアヒルがいて、秒針の動きに合わせてヒナが歩く。なんともいえないホノボノさを味わってちょーだいねん。

RUNさせて1分ほど待つと、「いまなん時？」と聞かれるから、現在の時刻を、時、分、秒(それぞれ12時間制)とカンマで区切って入力するだけ。スクリーンいっぱいにかわいいアナログ時計が動き出すぞ。ただし、処理があまりに遅くて、1時間に数秒ずつ遅れるのがタマに傷なんだよねえ。

でも、その辺がアナログ時計らしいと言えば、言えなくもない。

Windowsマシンと違って、MSX1にはカレンダーを保持するハードウェアがなかったのよねー。だから毎回起動時に時刻を問い合わせてきちゃうのだ。

電光板

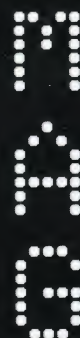
▶▶▶▶▶ DENKO.BAS

飲食店の看板やデパートなんかでよく見かける電光掲示板は、スマートにメッセージを伝える魔法の看板だ。電球やLEDを画素に見立てて、その集まりで文字を表示している。これを、MSXでやってしまうのがこのプログラムだ。

操作は簡単。プログラムをRUNさせたら、指示に従ってメッセージを入力するだけでいい。スクリーンに拡大された文字が次々に流れていく(キモチイイ〜)。

受け付けるメッセージの文字数は全部で200文字までだ。プログラムの仕組みは、入力された文字のドット構成をMID$関数とVPEEK命令を使って分解、拡大表示している。このプログラムでは、ASCII文字列にのみ対応している。っていうか、縦表示だと、ひらがなやカタカナを使っても、濁点、半濁点がうまく再現できないからね。

さあ、今年の冬は電光板を使って好きな人に告白してみよう！

メッセージが拡大表示でスクロールしていく様子は圧巻だ。これもWindowsマシンの画面表示モードを800×600に切り替えて大きく表示させるとグー！

MSXプログラム集

うるとらべえごま

▶▶ KOMA.BAS

下町の路地、遠くでなにやら楽しい声が聞こえる。歩みを進めるうちに、やがて声の主と正体が。漬物樽に麻布を張った小さな世界。そこには可愛いべえごまが2つ、子供たちの熱い視線を浴びていた。なんて、ノスタルジックな始まり方だけど、いわゆるべえごまなのだ。プログラムをロードしてRUNさせるだけ。これ、いちおうゲームなんだけど、鑑賞していて楽しいのでBGVでとりあげてみた。ルールは、「しろ」と「あか」に分かれたべえごまが戦って、ポイントがゼロになったほうが負けだ。こまが当たったとき、火花が散って本物以上？にリアルな音がするので、見ものだぞ。

で、プログラム解説だ。こまと火花はスプライトでデザインされている。音は、一瞬だけチャンネルを開いて閉じるという方法で、金属っぽい音にしている。

音量を上げて楽しもう！

ぶらっく・ボード

▶▶ KOKUBAN.BAS

グラフィックツールはどれも機能が複雑で難しい。そんなあなたのために、電子黒板を贈ろう。フリーハンドで、ラインを引く機能と、黒板消しの機能しかないけれど。

プログラムをRUNさせると、使いたいデバイスを聞いてくるので、好みの番号を押す。ただし、MSXPLAYerでは、マウスは機能しないので注意。画面左上に鉛筆と消しゴムのアイコンが現れて、中央にカーソルが表示される。これは、鉛筆なので、そのまま[スペース]キーやボタンを押しながら、好みの絵を描くことができる。絵を消したいときは、消しゴムのアイコンと重なるように、鉛筆カーソルを動かして、[スペース]キーかボタンを押すと、カーソルの形が消しゴムに変わる。絵の上をそのままなぞれば、絵はキレイに消えていくのだ。

伝言板にも使えるかも!?

あれっこわれたかな？

▶▶ KOWARETA.BAS

マシン語で、パソコンが壊れたように見せかけるプログラムだ。まず、友達にMSXPLAYerをひとしきり説明する。そうして、気づかれないようにプログラムをRUNさせたら、「適当にいじってていいよ」と言って、部屋を出る。友達は何も知らずにキーをたたき始めるだろう。そして、[スペース]キーを押した瞬間。「ピッ！」と音がして画面がメチャクチャになる。さらに、[CAPS]キーランプまで勝手に点灯してしまったりする。友達は真っ青になって慌てるんじゃないかな。

プログラムを停止させるには、[CTRL]＋[STOP]キーを押せばいい。このとき、[CAPS]キーのランプが点灯しているのに小文字しか入力できないときは、2回程度[CAPS]キーを押せば入力できるようになる。プログラムでランプを点灯させているためなので、驚かなくて大丈夫。

ラインアニマル

▶▶ LINE.BAS

大昔、ゲームセンターにあった「クイックス」というゲームを覚えているだろうか。これは、そのゲームの中で動く線でできた生物「クイちゃん」をキミのディスプレイ上に復活させようというもの。ただし、ゲームセンター版からは何年も経っているんで、かなり動きが鈍い。だからといって、変につっついたりせずにゆっくり老後を見守ってやってくれ。使い方は、ロードしてRUNするだけだ。

線の動く仕組みは、曲線を描いてくねくねと生き物のようにすばやく動かすために、あらかじめ円の座標を計算して、配列変数に代入しておく。はじめにラインを描いた後、そこからの相対位置と乱数で、新しい場所に違う色でラインを描いている。スクリーンの周囲に当たり判定があって、はみ出しそうになったら、今度は内側に切り返す。

クイちゃんをかわいがってね。

マシン語をBASICのDATA文にするプログラム

▶▶▶▶▶ SAVER.BAS

メモリ上のデータ(マシン語プログラムなど)をBASICのDATA文にするプログラムだ。簡単なプログラムだが、BASICとマシン語を使ったプログラム開発では、大きな威力を発揮することだろう。

使い方だが、まず、メモリ上にBASICにしたいマシン語データがあることが前提だ。プログラムRUNすると、開始アドレスと終了アドレス、そしてセーブするファイル名を聞いてくる。アドレスは16進数で入力しよう。それらをすべて入力し終わると、ただちにDATA文に直されファイルに出力される。出力されたファイルの冒頭には、ローダープログラムが付いていて、その後にDATAが続いている。また、出力ファイルをそのまま実行すると、データをメモリにロードして、データの先頭をマシン語コールする仕様になっているので、自分のプログラムに合わせて、変更を加えておこう。

BASICのプログラムだけでは処理が遅くなる部分はマシン語プログラムの出番だ。マシン語プログラムのBASIC化は意外と面倒だが、このツールなら簡単便利！

```
210   FOR B=A TO A+7
220     READ D$:D=VAL("&H"+D$)
230     IF D$="END" THEN A=USR(0):END
240     CS=CS+D:POKE B,D
250   NEXT
260   L=PEEK(&HF6A4)*256+PEEK(&HF6A3)
270   LOCATE 18,3:PRINT L
280   READ S$
290   CS$=RIGHT$("00"+HEX$(CS),3)
300   IF CS$<>S$ THEN GOTO 330
310   A=A+8
320  GOTO 200
330  '===== ERROR =================
340  PRINT L;"ｷﾞｮｳﾌｷﾝﾆ ﾆｭｳﾘｮｸﾐｽｶﾞ ｱﾘﾏｽ"
350  BEEP:END
360  '===== DATA ==================
1000 DATA 1B,2B,10,FA,C9,21,47,F8,379
1010 DATA 11,F2,2E,18,06,21,47,F8,2AF
1020 DATA 11,F3,2E,D5,11,F6,F7,3A,43F
1030 DATA 63,F6,FE,04,D0,11,F8,F7,52B
1040 DATA C9,79,B7,CA,71,2E,21,77,3FA
1050 DATA 2E,E5,CD,71,2E,79,C8,21,3E1

color   auto    goto    list    run
```

スプライトエディタ(MSX2対応)

▶▶▶▶▶ SPED2.BAS SPED2.LDR DATA.SP2

このプログラムは、簡単にスプライトモード2用のデータが作れるようにしたスプライトエディタだ。スプライトモード2の16×16ドットを、2枚重ねして使用するパターン専用のエディタになっている。

プログラムを実行すると、画面は次のようになる。一番左がカラー・ウィンドウ。やや真ん中よりにあるのがエディット・ウィンドウ。右がスプライト・ウィンドウだ。それぞれのウィンドウでの操作は、それぞれのモードに移行して行う。操作法は、コマンド表を参照してほしい。

作成したスプライトデータは、SPED2.LDRプログラム(ローダー)をつけて実行させることができる。サンプルデータとして、DATA.SP2を用意したので、参考にしてほしい。

スプライトを2枚重ねで使用すると、かなり表現力のあるパターンを作成することができる。スプライトエディタはプログラム作成に欠かせないツールだ。

表1 スプライト・エディットモード

キー	機能
[←][→][↑][↓]	ドットを指定
[スペース]	指定された色でドットをON/OFF
[リターン]	スプライト・セレクトモードへ
[M]	カラーセレクトモードへ
[Z]	1枚目のスプライト色を指定
[X]	2枚目のスプライト色を指定
[C]	1枚目と2枚目のORした色を指定
[R]	データをロード
[W]	データをセーブ
[D]	BASICのDATA文にしてセーブ
[SHIFT]+[CLS]	スプライトをクリア
[ESC]	終了

表2 カラー・セレクトモード

キー	機能
[←][→]	色を選択
[↑][↓]	スプライトの各ラインの指定
[スペース]	指定ラインに指定色をセット
[Z]	1枚目のスプライトを指定
[X]	2枚目のスプライトを指定
[M]/[リターン]/[ESC]	スプライト・エディットモードへ

表3 スプライト・セレクトモード

キー	機能
[←][→][↑][↓]	スプライトを指定
[スペース]	スプライトパターンをコピー
[リターン] / [ESC]	スプライト・エディットモードへ

キャラクタエディタ

▶▶▶▶▶ CHED.BAS, CHED.LDR, CHED.DAT

このプログラムは、MSXで使えるすべてのキャラクタ(256文字)を好きなように変更し、さらにBASICのDATA文にすることできるツールだ。

このエディタには、キャラクタセレクトモードと、キャラクタエディットモードがある。キャラクタセレクトモードで対象のキャラクタを選択し、キャラクタエディットモードでパターンを編集する。

両モード共通のコマンドと、それぞれのモードで使えるコマンドは表の通りだ。

DATA文形式で出力したデータをBASICで使用するには、ローダー、CHED.LDRを使用する。CHED.LDRをロードしておき、次にBASICのMERGE命令でデータファイルをマージする。実行したら、キーボードから文字を入力してみよう。うまくいったら、CHED.LDRとデータ文を自分のプログラムに取り込むといいだろう。ちなみに、ローダーを使ってパターンを設定したあとで、SCREEN文を実行すると、全部消えてしまうので気をつけて。

サンプルデータとして、CHED.DATが収録されているので、まずはエディタで読み込んでみて、参考にして欲しい。

表1 共通コマンド

キー	機能
[C]、[V]	文字色を変える
[B]、[N]	背景の色を変える
[R]	データをロード
[W]	データをセーブ
[D]	DATA文の形でセーブ
[ESC]	終了
[リターン]	他のモードに行く

表2 キャラクタエディットモード

キー	機能
[←][→][↑][↓]	ドット位置の指定
[スペース]	キャラクタのドットのON/OFF
[CLS]	キャラクタのクリア
[1~7]	ロール、反転、回転

表3 キャラクタセレクトモード

キー	機能
[←][→][↑][↓]	キャラクタの選択
[スペース]	キャラクタパターンのコピー、およびテスト・ウィンドウへのキャラクタ移動

キャラクタを変更してゲームキャラにするもよし、文字を変形して味のある画面作りをするもよし。いずれも1クラス上のプログラムを作成するのに役立つぞ!

MSXプログラム集

FM音色エディタ

▶▶▶▶▶ FMEDIT.BAS

MSXPLAYerに備わっているFM音源の音色エディタだ。

このプログラムには、3つの画面がある。まず、キャリアオペレータのパラメータ変更画面と、モジュレータオペレータのパラメータ変更画面の2つのモードがあり、実行直後は、モジュレータオペレータ変更画面になっている。現在のモードは、画面右上に「CAREER」/「MODULATOR」の文字で明示される。この2つのモードの切り替えは[ESC]キーで行う。両モードでの操作は表を参照してほしい。特に難しいことはないが、スライダーの値を変更するには、[スペース]キーを押して、指型カーソルが指を曲げている状態のまま上下の矢印キーを押す必要があるのでコツをつかんでおこう。

最後の1つは、音色の編集以外の各種操作のために用意されている画面だ。この画面では、[CAREER]/[MODULATOR]の各モードで[P]キーを押したときに演奏される音程や音の長さをMMLで指定できるため、PLAYデータ画面と呼ばれている。この画面での操作は、矢印キーでメニュー項目を移動し、[スペース]キーで選択する。あとはメッセージに従って、指示された事柄を入力すればいいぞ。

PLAYデータの設定以外に、音色データのセーブとロード、プリセットされた音色からのコピーなどが行えるようになっている。

FM音源は、パラメータの値と音色のイメージがなかなか結びつかないが、スライダーをあれこれ動かしながら試していけば、イメージに近づけることができるかも。

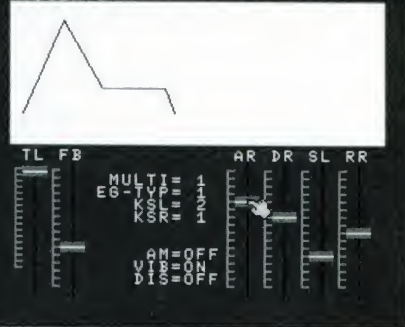



表　「CAREER」/「MODULATOR」両モードでの操作

キー	操作
[ESC]	キャリアオペレータとモジュレータオペレータの変更画面切替
[INS]	PLAYデータ画面へ切替
[スペース]+[↑][↓]	パラメータ変更(スライダー)
[スペース]	パラメータ変更(数値、ON/OFFタイプ)
[↑][↓][←][→]	パラメータ項目間移動
[P]	音色テスト
[CTRL]+[STOP]	プログラム終了

YJKモード用グラフィックエディタ

▶▶▶▶▶ GRP.BAS

SCREENモード10、11のグラフィックエディタだ。このツールはマウスで操作する。MSXPLAYerの「BASIC(マウス版)」を起動し、[F12]キーを押してマウスをMSXPLAYer側に切り替えて使おう。

このプログラムには、「YJK」と「RGB」の2つのモードがある。これはBASICのSCREEN10とSCREEN11の違いと同じ。プログラムをRUNさせた直後はYJKモードになっている。YJKモードでは、JKの値を2バイトにまたがって設定しなければならないため、1ピクセルは横4ドットになってしまう。

ウィンドウの操作は、いちばん右のタイルで色を選び、左に並ぶコマンドを選択して描画する。真ん中に並ぶ棒グラフは色を作成する場所だ。左からYが4つとJKが並んでいる。それぞれの列でX座標を合わせて、上側の+または下側の−をクリックすることによって、YJKの値を変更する。

左に並ぶ描画コマンドを選択するとウィンドウが閉じ、「Q」というアイコンのみが表示される。もういちどウィンドウを表示したければ、「Q」の上で右クリックすればいい。

このプログラムでSAVE、LOADコマンドを使うには、以下の1行を追加する必要がある(すまん)。

```
4325 IF ERL=3530 OR ERL=3680 THEN RESUME NEXT
```

表1　YJK/RGBモード共通

コマンド	機能
PSET(Ps)	1ピクセルの点を描く
LINE(Ln)	直線を描く
BOX(Bx)	四角を描く
BOXFILL(Bf)	内部を塗りつぶした四角を描く
FILL(Fl)	画面全体を塗りつぶす
COPY(Cp)	ドラッグ範囲をコピー
MODE(Md)	YJK/RGBモードの切替
SAVE(Sv)	データを保存
LOAD(Ld)	データを保存

表2　RGBモード独自の操作

コマンド	機能
PAINT(Pa)	BDR色を境界として塗りつぶす
CIRCLE1(C1)	ドラッグ範囲内に円を描く
CIRCLE2(C2)	ドラッグを半径とする円を描く

直前に行った描画の操作はすべて、マウスの右ボタンで取り消すことができる。SAVE、LOADはできないがSCREEN10の独特の描画モードは理解できるだろう。

MSX2専用自動ローダ／MSX1専用自動ローダ

▶▶▶▶▶ MSX2IPL.BAS, MSX1IPL.BAS

とても便利なファイラプログラムだ。画面に表示されている実行可能なファイルの一覧を選んで、[リターン]キーで実行できるすぐれものだ。

このプログラムはMSX用とMSX2用がある。MSX用は横40字モードで、MSX2用は横80字モードで動作する。MSX2でも横80字にすると漢字が読めない場合はMSX用を使って欲しい。

プログラムを実行すると、ディスクの中から拡張子が“.BAS”のファイルのみ一覧として表示される。

あとは、実行したいファイルにカーソルを合わせて[リターン]キーを押すだけ。これで、指定したプログラムがロードされ、実行される。このプログラムのカッコイイ使い方は、“AUTOEXEC.BAS”というファイル名を付けてセーブしておき、MSXをリセットしたときに一番初めに実行させることだ。

MSXPLAYerでは、MSX2専用自動ローダだけで事足りるが、文字が小さくなってしまい読みづらいこともある。そんなときはMSX1専用を使おう。

FM音色キーボード

▶▶▶▶▶ FMKEYBD.BAS

MSXのキーボードを使用して演奏できる簡易シンセサイザだ。

キーボードの割り当ては2段に分かれていて、対応は図の通りだ。同時に6声まで発生できるから、複数のキーを押して、和音を弾くことができるぞ。ただし、6個以上のキーが押された場合は、最後に押されたキーが無視される。

このキーボードには、音色を割り当てることができるが、このとき上下段にそれぞれ別の音色を割り当てることもできる。[INS]キーを押すと、上下同じ音色のモードと、上下に別々の音色を設定できるモードに切り替わる。音色の設定は、[ESC]キーで音色設定モードにしてから、好きな音色をカーソルキーで選んで決定する。

上下分割モードのときは、[リターン]キーで音色を決定したあと、割り当てを設定する。[←]で下段へ、[→]で上段へと割り振る。[リターン]キーで設定完了だ。

G# A#　C# D#　F# G# A#　C# D#
2 3　5 6　8 9 0　^ ¥
Q W E R T Y U I O P @ [
ソ ラ シ ド レ ミ ファ ソ ラ シ ド レ

C# D#　F# G# A#　C# D#　F#
S D　G H J　L ;　]
Z X C V B N M , . / _
ド レ ミ ファ ソ ラ シ ド レ ミ ファ

ミュージック・キーボード

▶▶ KEYBOAD.BAS

MSXの基本的なサウンドチップ(PSG)を使ったキーボードプログラムだ。パソコンのキーボードはもちろん、MSXPLAYer上のソフトウェアキーボードを使っても演奏することができる。

ただし、MSXPLAYer側の[CAPS]キーをオンにしておかないと演奏できないぞ。使用できるキーと対応する音階は次の通りだ。

F# G# A#　C# D#　F# G# A#　C# D#
1 2 3　5 6　8 9 0　^ ¥
TAB Q W E R T Y U I O P @ [RETURN
ソ ラ シ ド レ ミ ファ ソ ラ シ ド レ

MSX-BASICグラフィックワークブック

このコーナーのプログラムだけでは足りないアナタ。CD-ROMに収録されている「MSX-BASICグラフィックワークブック」のPDFファイル(MSXgwb.PDF)をのぞいてみてほしい。

これはアスキーから発行された書籍の一部を抜粋したもの。内容は、関数を使って描き出す目にも鮮やかなグラフィックスや、ゲーム作成のヒントにもなる楽しいグラフィックスを、短いリストで描くプログラムを集めたもの。リストを見ながら、プログラムを入力しなければならないけれど、動いたときの感動はひとしお。きっと、「よくこんな短いプログラムでこんなことができるものだ」と思ってもらえるのではないかな。

同PDFには、MSX用の画面レイアウト用紙も収録されているので、印刷して活用しよう！

MSXプログラム集

漢字テキストエディタ

▶▶▶▶▶ MKUSR.BAS、KED.BAS

漢字表示に対応したテキストエディタだ。ただし、MSXPLAYerにかな漢字変換FEPがないので、単漢字変換しか行えない。また、プリンタはサポートしていないので、印刷は行えない。しかし、漢字テキストファイルの作成や内容確認などには使える。

このプログラムは、マシン語のバイナリファイルを作成するプログラム“MKUSR.BAS”とエディタプログラム“KED.BAS”本体の2つに分かれる。MKUSR.BASと実行すると“kedusr.bin”というマシン語のバイナリファイルが作成され、セーブされる。一度バイナリファイルが作成されるとKED.BASが、このファイルを読みに行く。ただし、バイナリファイルと、KED.BASは、同じドライブに存在する必要がある。

このエディタで扱うことのできるテキストの量は、半角文字で約8000文字以内、そして行数は200行以内だ。なお、ここでの行数は実際の画面で表示されている行数のことで、ファイル中での改行の数ではない。

このプログラムには大きく分けて2つのモードがある。実際に文字を打ち込んでいくエディットモードとファイル操作などを行うコマンドモードだ。各モードでの操作は表のとおり。

表1　コマンドモードでの操作

[E]	エディットモードに入る
[P]	印刷（※使えない）
[S]	ファイルに書き込む
[L]	ファイルから読み込む
[Q]	エディタを終了する

表2　エディットモードでの操作

[CTRL]＋[E]、[↑]	カーソルを上に移動
[CTRL]＋[X]、[↓]	カーソルを下に移動
[CTRL]＋[D]、[→]	カーソルを右に移動
[CTRL]＋[S]、[←]	カーソルを左に移動
[CTRL]＋[W]	前ページに戻る
[CTRL]＋[Z]	次ページに進む
[CTRL]＋[R]、[INS]	挿入/上書きモードの切替
[CTRL]＋[M]、[リターン]	挿入モードでは行を2つに分割。上書モードでは次行の先頭に移動
[CTRL]＋[O]	行を2つに分割
[CTRL]＋[I]、[TAB]	挿入モードでは次のタブストップまで空白を挿入。上書きモードでは次のタブストップまで移動
[CTRL]＋[G]、[DEL]	カーソル位置の1文字を削除
[CTRL]＋[H]、[BS]	カーソルの左側の1文字を削除
[CTRL]＋[Y]	カーソルのある行を削除
[CTRL]＋[P]	削除された行を貼り付け
[CTRL]＋[L]、[HOME]	画面を書き直す
[ESC]	コマンドモードへ戻る

単漢字変換ならば漢字を入力することができる。また、ASCII形式でセーブしたBASICプログラムの編集に使うという手もある。

漢字スケジュールメモ

▶▶▶▶▶ SCHDUL.BAS

漢字テキストモード上で動作するスケジュール帳プログラムだ。日々のスケジュールを手帳の感覚で書き込み、内容をディスクに保存したり、スケジュールを照会したりできるようになっている。ただし、MSXPLAYerでは、プリンタサポートがないので印刷ができない。うーん、残念！

このプログラムを“RUN”させると、まずMSXの時計に設定されている日付を表示し、今日の日付を入力するよう促す。日付を変更したい場合は、西暦と月と日を2桁づつ（西暦は下2桁）を“/”（スラッシュ）で区切って入力し、[リターン]キーを押す。日付の入力が終わると、ディスクからスケジュールの保存されているファイル“schdu1.dat”を読み込んで、今日の予定を表示しコマンド待ち状態となる。最初に起動したときには、スケジュールのはいったファイルは存在しないので、枠だけを表示してコマンド待ち状態となる。使えるコマンドは表のようになっている。

[W]キーを押してスケジュールを書き込んでいこう。スケジュールが増えたら、ページ切替や、照会で目的のスケジュールを閲覧できる。

なお、単漢字変換による漢字の入力は「Tips集」を参照してほしい。

表　コマンド一覧

[←][→]	ページ切替
[↑][↓]	カーソル移動
[F1][W]	データの入力と修正
[F2][S]	日付によるデータの照会
[F3][I]	カーソル位置にデータを挿入
[F4][D]	カーソル位置のデータを削除
[F5][P]	印刷（※使えない）
[F6][E]	ファイルの出力と終了

市販のスケジュールソフトが複雑過ぎて難しいという人にオススメなシンプルさだ。子供用のスケジュール帳としても使えるかも!?

レイトレーシングプログラム

▶▶▶▶▶ TRACE.BAS, SAMPLE.BAS

CGを使った映画は、いまや全盛だけど、基本はすべてここにあるんだぜ。さあ、19000色を使った3DCG作成プログラムの紹介だ。プログラムを動かしてから画像ができるまでは、かな〜り時間がかかるけど、MSX2＋の機能を最大限に引き出すために作られたレイトレーシングアルゴリズムの威力をぜひぜひ、見てちょーだい。

まずは作品から見ちゃおう。SAMPLE.BASは、レイトレーシングプログラムで描いた作品を見るためのプログラムだ。実行すると、ファイル名の一覧を表示して、見たいファイル名を問い合わせてくる。ディスクにはSAMP7.YJKとSAMP8.YJKというサンプルがはいっているぞ。ファイル名を入力すると作品が見られるというわけだ。どう？ MSXで描いていても結構リアルでしょ。

さて、このようなCGを実際に計算しながら作成するのがTRACE.BASだ。レイトレーシングは、光が光源から物体に当たり反射して目に届くまでをシミュレートしながら1枚の絵を描いていく。この計算がとても時間がかかるのだ。実行して何十分か待っても画面の変化がほとんど何もないので、壊れたかと思うかもしれない。しかし、それぐらいでへこたれてはいけない。なにしろ、MSXPLAYerの速度が[NORMAL]だと、1枚のグラフィックを描くのに数日かかるのだ。え？ そんなに待っていられないって？ ならばMSXPLAYerの[∞]ボタンで最高速にしておこう。これなら1日か2日で描き終わる。いや、それでも1日や2日はかかるのだけれども。

さて、TRACE.BASは5000行以降のデータ文を書き換えることで、違ったグラフィックスを描ける。実はこのデータの書き方をちゃんと説明するには、それだけで十数ページを要するので無理（すまぬ）。その代わりと言ってはなんだが、先ほど眺めた作品、SAMP7.YJKとSAMP8.YJKのデータ文をディスクに収録している。これをTRACE.BASの5000行以降と置き換えることで、実際にSAMP7やSAMP8と同じ作品を描き出すことができるのだ。

方法は簡単。まずは、LOAD命令で、TRACE.BASをディスクから読み込む。

`load "trace.bas"` [リターン]

次に、データであるSAMP7.DATもしくはSAMP8.DATをMERGE命令で結合するのだ。

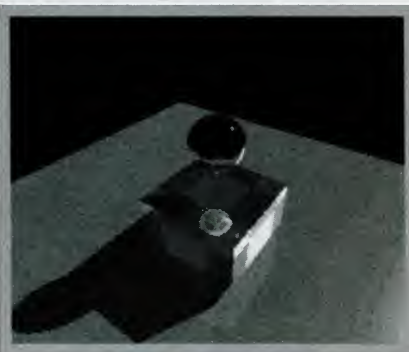
`merge "samp7.dat"` [リターン]

これで、5000行以降がSAMP7.DATに変更された。興味があればLIST命令で確認してみよう。

最後に結合されたプログラムをRUNコマンドで実行する。

`RUN` [リターン]

あとは気長に待つべし！

レイトレはリアルなCG作成手法の定番だ。本来大型のコンピュータで作成するものなので、MSXにはちと荷が重いが、それでもここまでできちゃうMSXってスゴイ！

MSX-DOSの開発ツール

BASIC版の仮想ディスクBには、スペシャル付録としてMSX-DOSが収録されている。起動方法は、MSXPLAYerをいったん起動しておき、仮想ディスクアイコンのBをクリックしてからMSXPLAYerの[RESET] ボタンをクリックする。すると、起動時に仮想ディスクBのMSX-DOSを読みにいき、見事DOSが起動するのだ。

MSX-DOSなんて何に使うんだ？という人はこの際ほっておいて、MSX-DOS感激〜という人には、さらに追加のオマケがあるのだ！

仮想ディスクBには、一部の間で要望の高い、MSX-DOS版のマシン語開発ツールも収録しているのだ。収録したツールはいずれもアスキー発行の書籍「MSX-DOSスーパーハンドブック」に掲載されたプログラムだ。

収録プログラムは表のとおり。エディタ、アセンブラ、デバッガがあればマシン語プログラムの開発は万全。さらに、漢字TYPEや、ファイルダンプ、ファイルコピーの便利ツールは、開発中の諸作業に必ずや役立つはず。

本書では、プログラミング解説がBASICまでとなっているので、マシン語そのものについては、どこにも解説していない。だから、これらのツールは、使える人にしか使えないということになってしまうけど、そこは勘弁してチョ！ その代わり、これらのツールの使い方については、作者の方々の了解を得て、書籍のマニュアルページをPDF形式でCD-ROMに収録している（MSX-DOSsh.PDF)。

```
MSX Self Assembler  Rev 1.4

Pass 1
No Error(s)

Pass 2
       0:                      ;HELLO
       1:   0100                       ORG
00100H
       2:
       3:   0009 =             STROUT  EQU
009H
       4:   0005 =             SYSTEM  EQU
00005H
       5:
       6:   0100 0E09                  LD
C,STROUT
       7:   0102 110901                LD
DE,MSG
       8:   0105 CD0500                CALL
SYSTEM
       9:   0108 C9                    RET
      10:
```

▶仮想ディスクBには、MSX-DOSでマシン語プログラムを開発するためのツールを収録したぞ。マシン語にチャレンジしたい人、昔とった杵柄な人はぜひ触ってみて！

表　ディスクイメージBに収録のプログラム

ファイル	内容
ASM.COM	アセンブラ（大貫広幸＋長井健一）
PED.COM	エディタ（張東一）
BUG.COM	デバッガ（清水真佐志）
FCOPY.COM	ファイルコピー（長井健一）
FCOPY2.COM	ファイルコピー2（長井健一）
FDUMP.COM	ファイルダンプ（張東一）
KTYPE.COM	漢字TYPE（長井健一）
HELLO.MAC	Helloコマンドソース（BITS）
CLS.MAC	CLSコマンドソース（BITS）

持ってるだけで幸せ！

超オススメ定番ツール

同人活動などのアマチュアイズムによって支えられてきたMSXでは、その中から優秀なツールがいくつも生まれているのだ。これらは今でもMSX用のデータを作るのには欠かせないものとなっている。また、MSXで作成したデータをWindowsなどで扱えるようにするためのコンバートツールなど揃えておくと便利。そこで、こうした「持っているだけで幸せな超オススメ定番ツール」を紹介しておこう。MSX環境用のソフトは基本的にMSXPLAYerでも利用可能だ。また、Windows環境用はWindowsで使用する。どのソフトの作者にも感謝!!

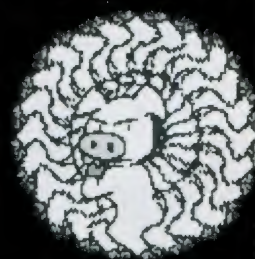
ゲーム開発用拡張BASIC

DM-SYSTEM2

- **環境** MSX-DOS
- **作者** GIGAMIX
- **配布** ギガミックス・ソフトウェア倉庫 http://www.zob.ne.jp/~goripon/MSX/SOFTWARE.HTM
- **配布ファイル名** DS2UTL-1.LZH、DS2UTL-2.LZH、DS2UTL-3.LZH、DS2UTL-4.LZH

GIGAMIX発行のディスクマガジン「ディスクメール」で使われていた拡張BASICと各種ユーティリティのセット。MSXにインストールすると、図形の回転COPY、音楽（BGM）演奏、PSG音源を使ったPCM再生、多彩な漢字表示、BPE方式という独自のメモリ圧縮／展開といった機能が、CALL xxxという形式でMSX-BASICから簡単に使えるようになる。DM-SYSTEM2は、特にゲーム制作で真価を発揮し、「マジカルラビリンス（本誌収録）」「クイズ！あたっちゃって25%」などでも存分に活用されている。詳細が知りたい人は、まずhttp://www.gigamixonline.com/ds2/に行ってみよう。使い方のABCがすべて記載されている。ここの内容は「DMシステム2・テクニカルハンドブック」という書籍としてGIGAMIXで発売中。

★ KPRINT TEST
Font Driver for Standard Kanji-ROM 1 & 2
6×6
愛のあるユニークで豊かな書体
8×8
愛のあるユニークで豊かな書体
10×10
愛のあるユニークで豊かな書体
12×12
愛のあるユニークで豊かな書体
14×14
愛のあるユニークで豊かな書体
16×16
愛のあるユニークで豊かな書体

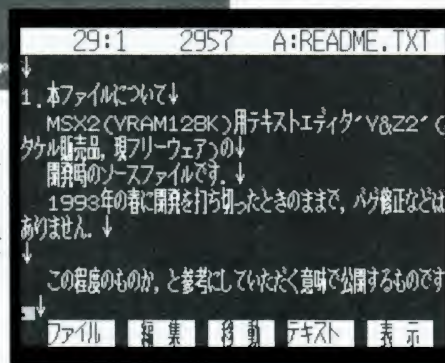
DM-SYSTEM2をインストールしておけば、70を超える拡張命令が使えるようになる。CALL VCOPY命令で画像をこのように回転させたり、CALL KPRINTで漢字表示も簡単だ。

MSX-DOS用テキストエディタ

Text editor V&Z

- **環境** MSX-DOS
- **作者** 宇佐美 仁織
- **配布** Tatsu's MSX情報局
http://gamelay.usami.com/~tatsu/
- **配布ファイル名** V&Z2SRC.LZH

最大8ファイルの同時編集、複数ファイルの同時表示、検索／置換／カット&ペースト、[BS]や[DEL]した文字の[CTRL]+[U]による無制限復活（アンドゥ）など、ともかく高機能がウリのテキストエディタ。ANK（英・数・カナ文字）専用の「Viper」と、日本語表示ができる「Zero」の2種類のエディタが同梱されている。配布ファイル内にはソースファイルも入っているため、どれが実行ファイルなのかわかりにくい。解説しておくと、Viperは「V.COM」があれば動作し「A>V ファイル名」で起動する。こちらは40文字モードでも80文字モードでも状態に合わせて表示される。一方のZeroは「Z.COM」があれば動作し「A>Z ファイル名」で起動する。やや表示される文字は大きいのだが日本語もしっかり表示される。

40文字モードでViperを起動してみたところ。主だった機能はファンクションキーに登録されていて、ファイルの保存、テキスト検索、テキストジャンプなど、一連の機能が完全に呼び出せる。プログラムのソース編集に便利。

日本語テキストをZeroに指定して起動してみたところ。MSXPLAYerには、単漢字変換機能が搭載されている（MSX-JEは含まれていないので注意）。

高機能マシン語モニタ

SHEM ver1.23

- **環境** MSX-DOS
- **作者** Sherry
- **配布** FSW倉庫（一般）
http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** SHEM123.PMA

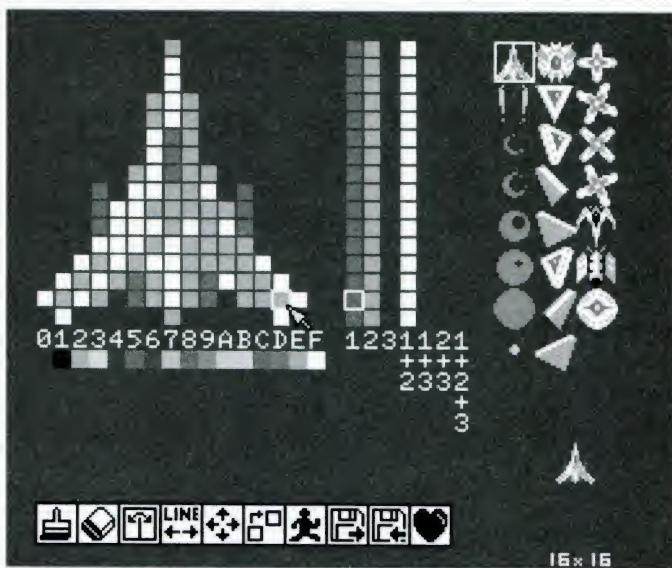
MSXマニア推奨のマシン語モニタ（CPUやメモリなどを覗くツール）。「A>SHEM」として起動するとメモリの16進ダンプ＋アスキー文字が表示される。ここでメモリの書き換えなどがサクサクできる。加えて、マシン語のディスアセンブル、レジスタ表示、スロットの切り換え、ディスクのセクタ単位でのアクセスなんてことができる。なお、本ソフトはPMA圧縮されて配布されているが、これはMSXでLZH普及以前に一般的だった形式。http://www.asahi-net.or.jp/~am9y-mn/fswlist.htmで配布されている「PMext」（CP/M用。MSX-DOSはCP/M互換なのでPMextはMSX上でも実行できるのだ）または「PMA」（MS-DOS用）で解凍できる。具体的には、それぞれ「A>PMEXT SHEM123 A:」「C>PMA E SHEM123.PMA」とすればよい。

操作は[CTRL]+○キーで行うものが多いので解説書をよく読もう。なお、MSX-DOSに戻る操作は[CTRL]+[ESC]だが、Windowsではスタートメニューが開いてしまうので、MSXPLAYerの仮想キーボードの[CTRL]キーをマウスで押しながら[ESC]キーを押してみて。

高機能スプライトエディタ

SEPT ver2.00

- **環境** MSX-DOS
- **作者** 夢神楽 遊睡
- **配布** FSW倉庫（グラフィック）
  http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** SEPT200.LZH

MSX2では複数枚のスプライトを重ねることで1ドットごとに色が付けられるが、この機能に対応したスプライトエディタが本ソフトだ。「A>SEPT」で起動する。標準的な16×16ドットサイズから複数枚のスプライトを組み合わせた64×64ドットサイズまでの編集ができる。スプライトの3枚合成モードがあり、ややこしくて面倒なスプライトのパターン作成が簡単に行える。作ったパターンはユーザーが選択することで、アニメーション再生ができるから、キャラの動きの確認もバッチリだ。操作がキーボードとマウスで行えるのが嬉しい。データの出力形式としては、BSAVE形式、BASICのDATA文形式、そしてマシン語ソースに組み入むためのDEFB（テキスト16進）形式が指定可能と至れり尽くせりだ。

1枚、2枚重ね、3枚重ねまでできるスプライトエディタ。ここまでできるソフトはSEPTぐらいではなかろうか。ただし、使いこなすにはスプライトの特性を熟知しておく必要があるのだ。

BMP形式に変換するツール

BMPセーバ

- **環境** MSX-DOS
- **作者** Kawasaki Yuusuke
- **配布** Ariakirasoft Download Page
  http://www.kawa.net/97/msx/download.html
- **配布ファイル名** BMPS111.LZH

MSX-DOS上で画像をBMP形式に変換するツール。MSX→Windows（Mac）への画像コンバートという点で同じコンセプトのソフトはいくつか存在するが、SCREEN5/7/8/10/11/12、そしてSCREEN7インターレス形式にも対応しているのはこれだけだ。また、ワイルドカード指定ができるため一括処理も問題ない。基本的にMSX-DOS上で「A>BMPS」としてから、あとはメッセージにしたがって[リターン]キーを押していけば変換できる。ただし、拡張子の1文字目がSで、3文字目が3/7/8/A/B/C/0のファイルしか受け付けない、などの制限があるため、必ずドキュメントを読んでからにすること。MSXの画像を直接Windowsで閲覧できるツールもあるのだが、MSX上で変換処理ができたほうがやっぱり助かることも多いのだ。

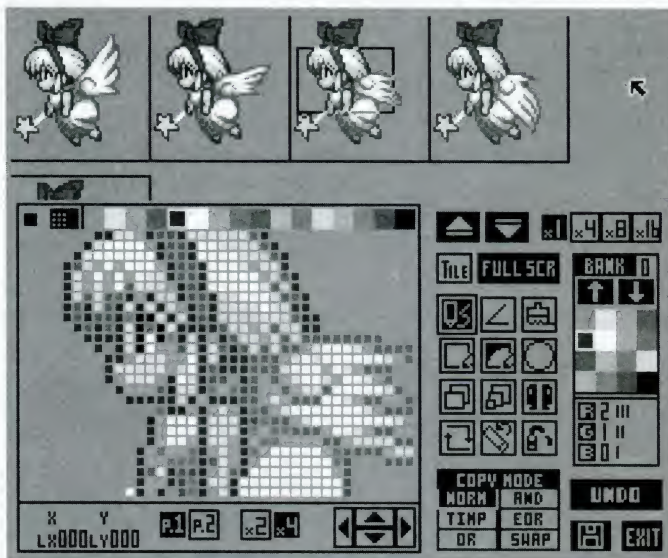
```
A>bmps
BMP saver  1.11b  Ariakirasoft

A:PICTURE.SC8 (RET:convert,C:cancel,P:pr
eview) ? converting ooooooo > OK

1 file(s) converted.  [Ariakirasoft 1994
 May]
A>
```

BMPSとだけ入力すると、自動的にファイルを探して、変換をするかどうかたずねてくれる。[リターン]キーで変換だ。また[P]キーを押せばその場でプレビューもできる。MSXで作った画像をWindowsで活かそう。

SCREEN5専用グラフィックツール

ANIMECHA

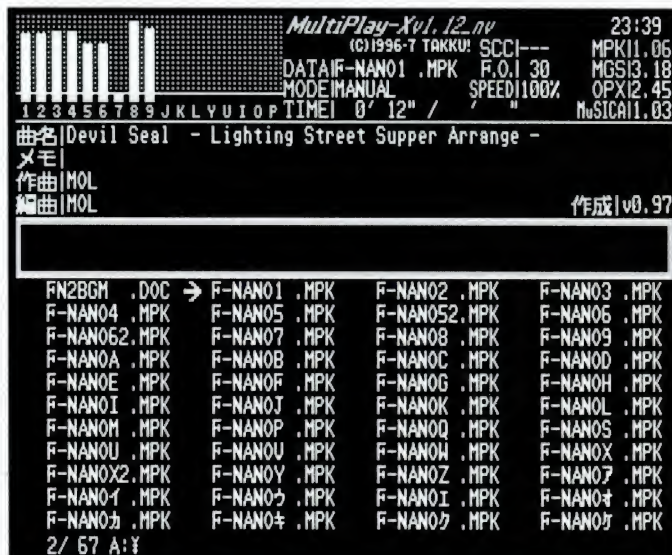
- **環境** MSX-DOS
- **作者** MAR'Z PROJECT
- **配布** TEMPEST
  http://www.tempest-j.com/msx/animecha.html
- **配布ファイル名** AM200.ZIP

名前通りグラフィックスによるアニメーションの編集を行うためのソフト。「A>AM」で起動する。これを使うと、アニメーションの各コマの描画、修正、そしてアニメーション再生ができる。仕様としては、最大99枚のセル（パターン）を重ね合わせて最大255のコマを作ることが可能だ。サンプルとして同梱されている「天使」のアニメーションを読み込んでアニメーション再生してみると、性能ばかりでなく、その楽しさも理解できるだろう。本ソフトはグラフィックツールとしても優れモノ。SCREEN5専用ではあるが、なかにはグラフサウルスより使いやすいと言う人もいるぐらいである。完全オンメモリ（起動時とデータのセーブ／ロード以外でディスクアクセスがない）で動作が軽快なのも涙モノ！

もともとアニメーションのために作成されただけあって、アニメーション編集にも優れている、メチャ便利なアニメカなのだ。

音楽データプレイヤー

MPX ver1.12

- **環境** MSX-DOS
- **作者** たっく！
- **配布** FSW倉庫（音楽）
  http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** MPX112.LZH

MSXで最終的に主流となったマルチプレイヤー。「A>MPX」で起動する。MGS／OPLLDriver(OPX)／MPK／MuSICAの計4種類のデータ形式の音楽ファイルが再生できる（ただし各ドライバを別途ダウンロードして、同じディレクトリに入れておく必要がある）。タイトル表示、トラック毎のON／OFF、ランダム再生、シャッフル再生など、基本的な演奏機能が充実していて動作も機敏。また、音楽データのファイル名と同名のテキストのファイルを作っておくと、再生時に自動表示してくれる機能なんかも備わっている（説明や歌詞などの表示に便利）。プレイヤーだけあっても音楽データがないと意味がないが、インターネットなどで検索すれば見つかるので、久しぶりにMSXサウンドを満喫してみちゃって。

画面下半分に音楽データが一覧表示されるので、これを選択すれば再生が始まる。FM音楽やPSG音源の懐かしき響きに酔いしれるのだ。

音楽データプレイヤー

MGSEL ver3.01v

- **環境** MSX-DOS
- **作者** Ain
- **配布** FSW倉庫（音楽）
http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** MGSELI01V.LZH

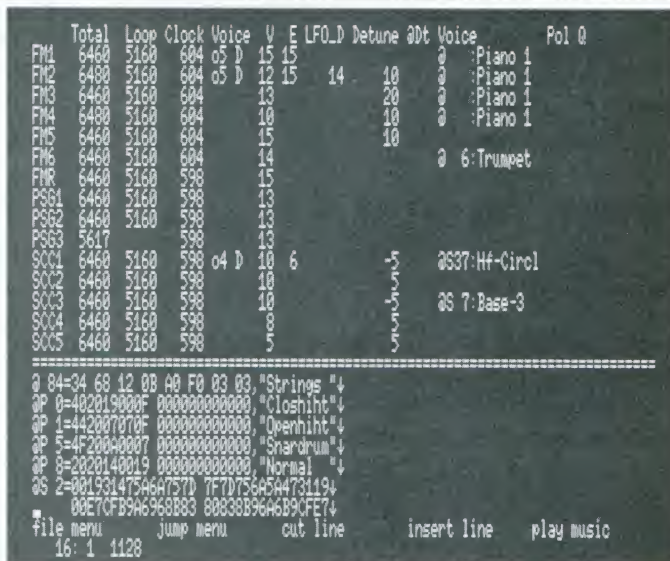
MGS形式の音楽ファイル専用プレイヤー。音楽プレイヤーとしては初期のものとなるが、見た目のデザインがイカしていたため、MGSの普及に一役買ったソフトとなっている。MGSELを使用するには、まずMGSDRV（v3.17以上）というドライバ（MGSDRV.COM）が必要である。また、MGSFというプログラムを用意し「A>MGSF MGSEL」としてタイトル一覧ファイル（MGSEL.DIR）を作成する必要もある。ちょっとメンドイが、このファイルがないとMGSELを実行したときに音楽データのリストが表示されないのだ。あとは「A>MGSEL」として起動し、表示されたタイトル一覧から演奏させたい音楽を選択するだけである。操作はキーボードのほかマウスも使える。演奏中は鍵盤やレベルメーターが賑やかでよろしい。

セッティングするまでが少しばかり大変だが、今からMGS形式の音楽データをインターネットでコレクションするのもオツなもの。今となっては、ややチープな音色がまたよかったりするのだ。

BGM作成ツール

MPK総合パック ver1.06

- **環境** MSX-DOS
- **作者** K-KAZ
- **配布** FSW倉庫（音楽）
http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** MPK106.LZH

同人ソフト「3D CAR ACTION F-nano2」で使用するBGMを作るために、いちから作られた音楽エディタ＋ドライバ＋技術資料からなるセット。MPKはMSXでは主流だったMML（Music Macro Language）で作曲するツールだ。「MuSICA」と同様にテキストエディタとコンパイラが一体となっているため、エディット→コンパイル→演奏という一連の作業がすべて行える。演奏しながらのエディットが可能なこともあり、確認しながらどんどん編集していけるのが便利。ドライバ部分の技術資料が公開されているので、アセンブラが理解できるなら自作プログラムに組み込むことも問題ない。なお、MPK形式の音楽ファイルは、別項の「MPX」でも演奏可能。FM／PSG音源に対応している。

現在のWindowsの音符入力が可能なシーケンサなどと比べると、数値入力的なMMLはかなり原始的な感じ。しかし、やってみると意外と馴染めるものだったりする。また音符が読めない人でも数値なら問題なしだ。

MSX-DOS用のCコンパイラ
HI-TECH C V3.09

- **環境** MSX-DOS
- **作者** HI-TECH Software
- **配布** FSW倉庫（一般）
http://www.interq.or.jp/white/white/fsw99/findex.html
- **配布ファイル名** HITECH-1.PMA

ANSI規格標準のCコンパイラだ。これを使うと、MSX-DOSのプログラムをC言語で記述し、開発することができる。C言語というと、MSX-C（アスキー）を使うのが標準的な選択だが、今や入手が難しい。その点、HI-TECH Cは今でも入手可能だ。プログラムを作って売ることも許されているのが嬉しい。マニュアルは英語だが、こちらについてもヘビーユーザーの手により、情報はリソースキットとして残してくれているから、これを見れば多くの問題は解決できる。HI-TECH Cはコンパイルが遅いという欠点があるが、これについてもMSXPLAYerをFASTモードにすれば、なかなか高速で快適になるぞ。

インストール方法

HI-TECH Cを使うにあたり、もっとも面倒なのがインストール。もともとCP/Mというi8080/Z80CPU用のOSであり、MSX-DOSで使うにはパッチ（修正）を当てる必要があるのだが、これが大変。試しにMSXPLAYer（MSX-DOS）を使ってインストールしてみたので、その手順を紹介しておこう。

まず、FSW倉庫からHITECH-1.PMAのほかに、「MSXで使うためのパッチ」（HPATCHES.PMA）と、「リンカ」（LINK.LZH）をダウンロードする。また、同じくFSW倉庫から「HI-TECH C Resource kit #1 ver1.51」（RESKIT1.LZH）をダウンロードする。これらは、すべて解凍しておくことをお忘れなく。また、あらかじめ2DDのフロッピーディスクを2枚用意してフォーマットしておこう。WindowsXPでは2DDのフォーマットができないので、Windows2000/Meなどで行うこと。その1枚目のフロッピーディスクに、HITECH-1.PMAの中のC.COMを除いた*.COMファイルをすべて、HPATCHES.PMAの中のファイルすべて、LINK.LZHの中のLINK.COM（上書き）、RESKIT1.LZHの中のPATCH1.BATを転送する。

パッチは2段階で当てることになる。まず、BASIC版MSXPLAYerを起動し、ディスクBで再起動してMSX-DOSを起動する。ディスクBから実ディスク（FDD）に切り替えて「A>PATCH1」として実行。これでHI-TECH Cの各ファイルにパッチがくっつく。終わったら一度MSXPLAYerは終了しておくこと。次に、RESKIT1.LZHの中のEZINST.LZHを解凍して、同じフロッピーディスクに転送する。再度MSXPLAYerを起動して、先ほどと同様にMSX-DOSを立ち上げ、実ディスク（FDD）に切り替える。そこで「A>EZPATCH」として実行。これでパッチが有効になる。

最後に、HITECH-1.PMAの展開ファイルの中に、パッチを当てたCGEN.COM/CPP.COM/CREF.COM/DEBUG.COM/LIBR.COM/LINK.COM/OBJTOHEX.COM/OPTIM.COM/P1.COM/ZAS.COMを書き戻す。それから、HITECH-1関係の全ファイルを2枚目のフロッピーディスクに転送してインストールは完了。同じディスクにC言語のソースファイルを用意し（コンパイル時に余計なエラーを出さないために、ファルの最後に1AH（EOF）のコードを入れておきたい）、MSX-DOSを起動して「A>C TEST」（TEST.Cの場合）とすればコンパイル開始。終了すればTEST.COMができている。これは感動だ。

```
MSX-DOS version 1.03
Copyright 1984 by Microsoft

COMMAND version 1.11

A>patch1
A>rem batchfile to concatenate patch pro
grams under MSXDOS-1
A>rem ONLY RUN THIS ONCE!
A>rem the original programs are saved as
 .BAK
A>rem you must still add the jump instru
ction and change the
A>rem exit status address (see documenta
tion)
A>COPY /B CGEN.COM+CGEN1.COM CGEN2.COM
        1 file copied
A>REN CGEN.COM CGEN.BAK
A>REN CGEN2.COM CGEN.COM
A>COPY /B CPP.COM+CPP1.COM CPP2.COM
        1 file copied
A>REN CPP.COM CPP.BAK
```

パッチを当てるには「パッチをくっつける」と「パッチを有効にする」という2段階の作業が必要だ。これは「A>PATCH1」を実行して、1段階目の処理をしているところ。

```
A>Ayaka CPP.com CPP.CFG
Customizer Ayaka Version 1.04 for MSX-DO
S1/2 By Tatsuhiko Syoji
Target file:CPP.com
Configuration file:CPP.CFG
Warning:This program isn't compatible fo
r XMARK family.(1st header)
Customize completed.

A>Ayaka CREF.com CREF.CFG
Customizer Ayaka Version 1.04 for MSX-DO
S1/2 By Tatsuhiko Syoji
Target file:CREF.com
Configuration file:CREF.CFG
Warning:This program isn't compatible fo
r XMARK family.(1st header)
Customize completed.

A>Ayaka DEBUG.com DEBUG.CFG
Customizer Ayaka Version 1.04 for MSX-DO
S1/2 By Tatsuhiko Syoji
Target file:DEBUG.com
Configuration file:DEBUG.CFG
█
```

こちらは2段階目の処理として「A>EZPATCH」を実行し、パッチを有効にしているところ。これさえ終わればインストールはほぼ完了だ。

```
MSX-DOS version 1.03
Copyright 1984 by Microsoft

COMMAND version 1.11

A>c test
HI-TECH C COMPILER (MSXDOS) V3.09
Copyright (C) 1993 HI-TECH SOFTWARE

Executing $EXEC

A>$EXEC
A>CPP -DCPM -DHI_TECH_C -Dz80 -I TEST.C
$C1.T
HI-TECH C PRE-PROCESSOR v3.10 (MSXDOS)

A>P1 $C1.T $C2.T $C3.T
HI-TECH C PASS1 v3.10 (MSXDOS)

A>CGEN $C2.T $C1.T
HI-TECH C CODE GENERATOR v3.10 (MSXDOS)
█
```

HI-TECH Cでコンパイルしているところ。C言語でMSX-DOS用のプログラムがガシガシ作れるってのはいいもんだ。プログラミングのお勉強にも使えるってもんよ。

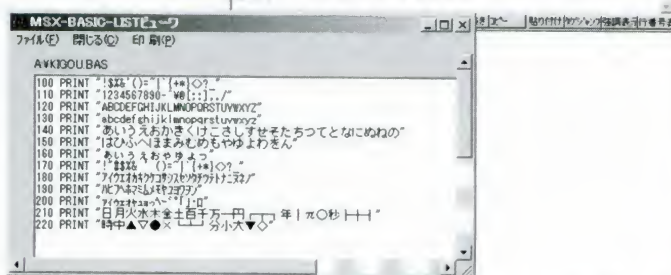
※PMAの解凍方法は「SHEM ver1.23」の解説を参照すること。
※Windowsから2DDフロッピーディスクにファイルをコピーしているときエラーが出た場合には、MSXPLAYerが起動していたらいったん終了し、Windowsでディスクチェックをしてみてほしい。
※インストールについては、RESKIT1.LZHの中のINSTHC.TXTというファイルに詳しい情報が載っているので、じっくり読み込んでみて。

MSXのプログラムリストをWindowsで表示

MSX-BASIC-LISTビューワ

●環境　Windows
●作者　上羽紘吉
●配布　ベクター
http://www.vector.co.jp/soft/win95/prog/se122526.html
●配布ファイル名　MSXBASIC.LZH

MSX-BASICの中間言語で保存されたプログラムをWindowsで表示するためのソフト。MSX-BASICでプログラムを「SAVE "ファイル名"」でセーブすると中間言語の形で保存される。これをWindowsのテキストエディタで開いても、謎の文字が表示されるだけである。Windowsで表示するには「SAVE"ファイル名",A」（アスキー形式を指定）とすればいいが、再保存は面倒なときはこれが便利。ちなみに、MSX特有の半角文字（ひらがな、罫線、記号類）については、他の全角文字や記号に置き換えてくれるのだ。MSXPLAYerはプリンタに対応していないし、WindowsにMSX用のプリンタは接続できないので、実は本ソフトは、MSXのプログラムをWindowsで印刷したいとき、もっとも役立つと思う。使ってみて。

MSX-BASICのプログラムをアスキーセーブしてWindowsのエディタで表示すると、MSX特有の文字が化けてしまったりする（アスキーセーブしなければ、もっと悲惨である）。

MSX-BASIC-LISTビューワを使えばMSXで普通にセーブしたプログラムを、Windowsでも読める文字に変換してくれるのだ。スペード、クラブ、ハート、ダイヤといったWindowsにはないマークも、▼、▲、▽、◇と何げにうまい記号に置き換えられる。ともかくMSXのプログラムをWindowsで印刷したいときには大変便利。

BSAVE画像対応Susieプラグイン

Susie32 Plug-in for MSX BSAVE File

●環境　Windows
●作者　市岡 耕平
●配布　ベクター
http://www.vector.co.jp/soft/win95/art/se059061.html
●配布ファイル名　PMSX010.lzh

MSXでBSAVEした画像はWindowsで直接表示することができない。Windowsで表示する方法の1つは、すでに紹介している「BMPセーバ」などで変換することだが、同様にWindows側で変換する方法もある。本ソフトは、Windows用の画像ビューワとして有名な「Susie」のためのMSX画像表示プラグインである。Susieがインストールされているフォルダに入れておくことで、BSAVE形式の画像ファイルと、グラフサウルス形式のファイルを表示できるのだ。もちろん、Susieの拡大／縮小／回転などの機能を使うことも問題ない。なお、BSAVE形式はファイル内部にSCREENモードの番号を記録しておらず拡張子のみで判別している。もし表示がうまくされないときは拡張子を確認してみること。

グラフサウルスver2.1のサンプル画像を、本プラグインをインストールしたSusieで表示してみたところ。Susieのフォルダに解凍したファイル（ifmsx.spi）を入れておくだけだ。対応しているMSXのSCREENモードは5/6/7/8/10/12となっている。

BMP画像をMSXのBSAVE画像に変換
BMP2MSX v1.11008

- **環境**　Windows
- **作者**　HRA!
- **配布**　BMPtoMSX
http://www.imasy.or.jp/~hra/BMP2MSX/
- **配布ファイル名**　BMPMSX_11008_J.LZH

BMPファイルをMSXの各種SCREENモードの画像に変換する。Windows上にあるデジカメ画像やビットマップ画像などを、このツールを用いれば何でもMSXにもっていける。出力形式として指定できるSCREENモードは2から12（9は存在しないため除く）で、インターレースも指定可能である。さらに、SCREEN2や4の、8ドット中に2色しか使えないモードを指定しても、かなりいい感じで変換される。その他、パレット番号の使用／不使用の選択や、指定したパレットセットの使用などにも柔軟に対応してくれる。対応画像はBMPだけなので、GIFやJPEG画像の直接指定はできないが、PaintShop Proなどで変換しておけばいいだけのこと。間違いなくMSXのプログラムを開発するのには必要不可欠だ。

デジカメの画像も、BMPtoMSXにドラッグ&ドロップすれば、MSXのデータに変換してくれる。簡単でしかも便利だ。

MSXの音楽演奏用Winamp用プラグイン
MSXplug

- **環境**　Windows
- **作者**　okazaki@angel.ne.jp
- **配布**　PC Sound Antiques
http://www.angel.ne.jp/~okazaki/ym2413/
- **配布ファイル名**　IN_MSX022.ZIP

Windowsの音楽プレイヤーとして有名な「Winamp」で、MSX用音楽データ（KSS／MGS／BGM／OPX／MPK形式）を直接鳴らすことができるようにするInputプラグイン。現在配布中のWinamp3ではWinamp2用のInputプラグインが直接利用できない。使えるようにするには「Winamp v2.x input plugin support」（http://www.blorp.com/~peter/の「Winamp v3.x plug-ins」メニューで入手できる）をインストールする。Winamp3のPREFERENCESダイアログに「Winamp v2.x input plugin support」が現れるので、［INSTALL］ボタンを押し、「in_msx.dll」（配布ファイルを解凍すると作られる）を登録しよう。やや動作が不安定な感じもするが、MSXで作られた音楽が、Windowsで手軽に再生できるのはとてもありがたい。

対応する音楽データをWinampにドラッグ&ドロップすれば演奏が開始する。うーん、MSX風味の音楽はえ〜なぁ。MSXplugのConfig用ダイアログボックスでは、音源ごとの音量調節なども行えちゃうよ。FM／PSG音源に対応だ。

▶MSXPLAYer Tips集

MSXPLAYerを使い込んでいくと、ちょっと疑問に思う点や困った点もでてくるだろう。ここでは、そういったMSXPLAYerの使いこなしのTipsをQ&A形式でまとめておく。基本的な使い方は、「スタートMSXPLAYer」にあるので、そちらを参照してほしい。

>>MSXPLAYerの操作と動作

Q 「SPEED」のボタンが「NORMAL」の状態でも、動作が異常に遅いのですが？

「Windows Update」、またはハードウェアの製造元のWebサイトから、最新のドライバをダウンロードして更新することで、改善される場合があります。

Q プログラムの実行中に「SPEED」のボタンを押しても大丈夫？

[SLOW][NORMAL][FAST]は、内部でのタイミングを維持したままスピードの制御を行っているため、プログラムの実行中に切り替えても大丈夫です。BASICソフトなどで初期設定の待ち時間を減らしたい場合などに使用してください。ただし、VDP(画面描画)の速度やBGMの速度は変化しません。ただし、ソフトによっては(特にゲーム)[NORMAL]以外だと正常に動作しなくなる場合があります。[∞]は、マシンスペックを最大限に引き出しているために、タイミングの制御がなされていません。ゲームなどは暴走する可能性が高いため、利用は限定されます。

SPEED
SLOW NORMAL FAST ∞

⬆「SPEED」の4つのボタン

Q プログラムの実行中に「FRAME RATE」のボタンを押しても大丈夫？

内部の処理は変わらないので問題は起きません。ただし、[HIGH]や[REAL]を設定した場合は、PCのスペックによっては全体的な動作速度が遅くなることがあります。

FRAME RATE
LOW MIDDLE HIGH REAL

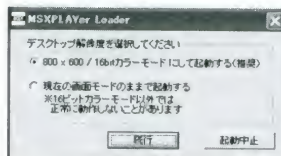
⬆「FRAME RATE」の4つのボタン

Q MSX-DOSを使いたいんですが？

MSXPLAYerを起動したら、ディスクアイコンの「B」を選択して、リセットボタンを押すと、MSX-DOSが起動します。

Q MSXPLAYerの動作速度が遅い、または他のアプリの速度が遅くなるのですが？

「SPEED」のボタンが[SLOW]になっている場合は、[NORMAL]にします。それでも遅い場合は、CPU速度が不足していると思われるので、「FRAME RATE」のボタンを1つずつ[LOW]側に変えて試してください。また、グラフィックチップのドライバを最新版にすることで改善されることもあります。各メーカーのWebサイトなどをご覧ください。なお、MSXPLAYerの起動時に「現在の画面モードのままで起動する」を選択した場合は、バックグラウンド側のアプリが重くなることもあるので、できるだけ「推奨モード」で起動してください。

起動時画面➡

Q 「マウス版」「ジョイパッド版」の違いって何？

ポート1にマウスが繋がっているか、ジョイパッドが繋がっているかの違いです。マウスを使う必要があるときだけ「マウス版」を起動してください。また、「マウス版」と「ジョイパッド版」がアクセスする仮想フロッピーディスクは共通ですので、ジョイパッド版で作ったファイルは、マウス版でもそのまま読み込めます。

Q MSXPLAYerは同時起動できる？

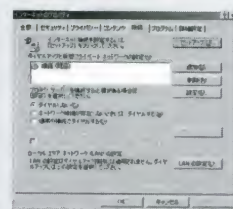
できません。別のゲームを遊ぶ場合は、起動中のMSXPLAYerを終了してから再度実行してください。

Q Windowsのエディタで作ったプログラムを、カット&ペーストでMSXPLAYerに移せるの？

移せません。MSXとWindowsの使用する文字データが異なるため、できないようになっています。

重要

Q MSXPLAYerでインストールができないのですが？

•アナログ電話回線でインターネット接続している場合は、インストール中に自動的に接続が開始されてしまうことがあります。[スタート]→[設定]→[コントロールパネル]→[インターネットオプション]→[接続]を開いて「ダイヤルしない」に設定してください。インストール終了後はもとに戻してください。

•オートランで起動するメニュー上からMSXPLAYerをインストールする際、InternetExplorerの設定によっては、ファイルのダウンロードの確認画面が出ます。[開く]ボタンをクリックするか、「このプログラムを上記の場所から実行する」をチェックしたうえで[OK]をクリックしてください。
詳細は公式ホームページのFAQをご覧ください。
http://www.ascii.co.jp/pb/ant/msx/faq.htm

Q タイトルバーの最小化ボタンを押したら、タスクバーが表示されなくなった！

「推奨モード」で起動した際、このような症状が起こる場合があります。MSXPLAYerに戻る場合は、[ALT]＋[TAB]キーなどを使用してください。

Q MSXとWindowsで同じファイルを同時アクセスするのは危険？

MSXPLAYerとWindows(エクスプローラ)から同時にフロッピーディスクへアクセスするのは、データを破壊する恐れがあるので危険です。MSXPLAYerの実行時には、可能な限りエクスプローラは使用しないようにしてください。ハードディスク内の仮想フロッピーディスクを使う場合は、同時アクセスは発生しません。

>>MSX実機との比較

Q MSXPLAYerの再現性はどれくらい?

十分なCPUスピードがあれば、MSX2+相当をほぼ完全に再現できます。また、オプション扱いの第二水準漢字ROM／FM音源(MSX-MUSIC)も搭載しています。ただしMSX-JEは搭載していないので、単漢字変換のみが可能です(右の項目を参照)。

Q 漢字フォントが、昔使っていたMSXと違うような?

漢字フォントのデザインはメーカーによって違いがありました。MSXPLAYerでは、JIS標準16ドットフォントを基にしたオリジナルフォントを使用しています。ちなみにBASICのフォントはどのスキンでも共通です。

Q 画面モードはどこまでサポートしているの?

すべてのスクリーンモード(SCREEN0～8、10～12)をサポートしています。

SCREEN9は、韓国のMSX2にのみ搭載されたハングル表示専用モードで、国内のMSXと同様にMSXPLAYerでは搭載していません。

Q MSXPLAYerで未対応のBASIC命令を教えてください!

①カセットテープ関係

MOTOR(エラーにはならない)、CLOAD／CSAVE(Device I/O errorになる)、カセットテープへのデバイス指定(BLOAD "CAS:",Rなど)には対応していません。

②プリンタ関係

LLIST、LPRINT、プリンタへのデバイス指定(MSX-DOSのデバイスファイルPRNなど)には対応していません(エラーにはならない)。

Q なぜMSX-JEがないの?

漢字BASICのソフトは、MSX-JEの種類によってフリーエリアの減少量が違います。そのため、特に松下のMSX2+で作ったソフトがSONYのMSX2+で動作しない場合がありました。その回避のために今回はMSX-JEを外しています。

ただし、単漢字変換はできます。方法は以下の通りです。

❶BASICを起動し、「call kanji」と入力する。
❷漢字モードになるので、[CTRL]＋[space]を押す。
❸ソフトウェアキーボードの「かな」をクリックする。
❹文字を入力すると、画面下部に変換候補が表示される。

Q ユーザーメモリ(フリーエリア)はもっと増えないの?

[CTRL]キーを押しながら起動(リセット)すると、仮想ドライブが切り離され、フリーエリアが増えます。ただしこの場合、仮想ドライブ(B:)は使えなくなります。また、起動時に[SHIFT]キーを押しておくと、DISK BASICが切り離されるため、フリーエリアをさらに増やすことができますが、この場合はフロッピーディスクが使えなくなります。これは実機のMSXではカセットテープ専用のソフトを動作させる場合に使用する機能としてあったものです。

Q スプライトの横並び制限(スクリーン3以下は4枚、それ以上は8枚)はある?

あります。MSXPLAYerではMSXを忠実に再現しています。また、その制限を演出として使っているプログラムもあるので、実機どおりの仕様としています。

Q SET PASSWORD、SET TITLEは使えるの?

使えません。保存されないようになっています。

Q MSXPLAYerから印刷は可能?

できません。MSXとWindowsで文字コードが異なるためです。

Q MSXPLAYerのカレンダーや時刻はどうなっているの?

Windowsのカレンダー・時刻と同期していますので、設定の必要はありません。

>>ファイル操作

Q ファイルはいくつまで作れるの? また、階層ディレクトリ(フォルダ)は作れますか?

ファイルはフロッピーディスク1枚に対して112個まで作れます。

階層ディレクトリ(フォルダ)は作れません。将来搭載予定のMSX-DOS2では対応しています。

Q 仮想フロッピーディスク機能って何?

ハードディスク内に存在するフロッピーディスクを「仮想フロッピーディスク」と呼びます。

MSXPLAYerに接続されているフロッピーディスクは1台ですが、ハードディスク内に4枚の仮想フロッピーディスクを持ち、それらを切り替えて使用することができます。

⬆ハードディスク内にある4枚の仮想フロッピーディスク

Q フロッピーディスクをMSXPLAYerで使ったら、MSX実機で動作しなくなった！

Windowsによっては、読み込みをするだけでもディスクの一部(ブートセクタ)のデータを書き換えてしまいます。その際にわずかですが起動用のプログラムを書き潰されてしまうソフトがあります。その場合はMSX実機でも起動できなくなり、元に戻すことも不可能です。必ずライトプロテクトノッチを開けて(=書き込み禁止にして)使用してください。

ディスクに直接セーブするタイプのゲームや、プロテクトノッチを閉じていないと起動できないタイプのゲームは絶対に使用しないでください。ユーザーディスクを別に用意するタイプのものは、ユーザーディスクのみを書き込み可能とすることで遊ぶことができます。

3.5インチフロッピーディスクノッチを開けると「書き込み禁止」になる。

ライトプロテクトノッチ➡

Q ファイルのコピーはMSX-DOSが楽って本当？

大量のファイルをコピーする場合、MSX-DOSのCOPYコマンドを使うと、ディスクの入れ替えの回数が少なくなります。BASICと少し違い、「COPY *.* B:」のように指示します(書式はMS-DOSとほぼ同じ)。

Q フロッピーディスクドライブからファイルをコピーして、昔作ったゲームを遊びたい！

MSX-BASICで、FDDから仮想フロッピーディスクにファイルをコピーする方法は、以下の通りです。

❶フロッピーディスクを書き込み禁止(ライトプロテクトノッチを開ける)にして、ドライブに挿入する。

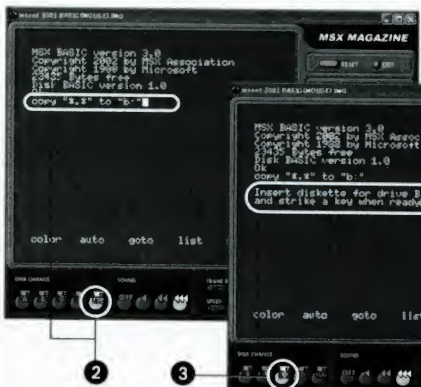
❷ディスクアイコンを「FDD」にして「COPY "＊.＊" to "B:"」と入力し、RETURN(Enter)キーを押す。

このとき、今選んでいるディスクアイコンが「A:」になる(ディスクアイコンの「A、B、C、D」とは関係ない)。

❸「Insert diskette for drive B: and strike a key when ready」と表示されたら、コピーしたいディスクアイコンに切り替えてRETURN(Enter)キーを押す。

このとき、仮想フロッピーディスク「C」と「D」は最初空の状態なので、このどちらかがよい。

❹これを繰り返すことでコピーできる。

ディスクアイコンの指定を間違えないように注意しよう。

Q 仮想フロッピーディスクとして使えるのは4枚まで？

今回のバージョンでは4枚となっています。機能的にはもっと増やすことも可能ですので、ご要望をMSXアソシエーションまでお寄せください。

>>手持ちのゲームで遊ぶ

Q 手持ちのゲームが操作できない、または勝手にカーソルが動いちゃう！

「マウス版」を起動していませんか？ 一部のソフトは(MSXに)マウスが接続されていることで誤作動する場合があるので、「ジョイパッド版」を使いましょう。

MSXはジョイパッドとマウスを同じコネクタに接続するため、このような症状が起こりえます。

Q ディスクのセクタを書き換えるようなプログラムでも問題なく実行できる？

できます。ただし、市販のゲームでディスクに直接セーブするタイプは、ディスクをWindows側で書き換える恐れがあるので、必ず書き込み禁止にして使用してください。

Q 手持ちのフロッピーディスクのゲームで遊ぶにはどうするの？

必ずフロッピーディスクを書き込み禁止(ライトプロテクトノッチを開ける)にし、ドライブに挿入してから、MSXPLAYer上のディスクドライブのスイッチを「FDD」に設定します。その状態で、MSXPLAYerのリセットボタンを押してください。また、フロッピーディスクドライブの機種によっては、Windowsの仕様上、MSXPLAYerから認識できないこともあります。

Q 昔のソフトを使ってみたいけど、動かないソフトってある？

①カセットテープ版とカートリッジ版のゲームソフト

プログラムを読み込む方法がないため動かせません。カセットテープは、MSX実機でフロッピーディスクに移すことができれば実行できます。フロッピーディスク版のBASICソフトならばたいてい動作します。

②1DDのディスク(360KB)フォーマットに記録されているソフト

Windowsのフロッピーディスクドライブが1DDに非対応であるため動かせません(Disk I/O Errorになる)。

③プロテクトがかかっているソフト

ソフトによっては不正コピー防止のために特殊なプロテクトがかけられ、動作しない、もしくはゲームが途中で停止するなど、異常な動作をする場合があります。

>>フロッピーディスク／ドライブ

Q 自作機ですが、フロッピーディスクドライブは何台まで接続可能？

一般的なMSXの構成と同じで1台です。MSXPLAYerでは、Windowsマシンに接続した台数に関係なく1台のみ使用できます。

Q BASICからフロッピーディスクが使えない？ エラーが出ます！

以下の理由が考えられます。
①MSXPLAYer起動時に[SHIFT]キーを押しておくと、フロッピーディスクは使えません。
②1DDフォーマットのフロッピーディスク(「MSX-DOS TOOLS」など)は、Windows側の仕様により読み込むことはできません。
③2HDフォーマットのフロッピーディスクは、MSXの仕様により読み込むことはできません。

Q フロッピーディスクを入れて起動すると止まります！

以下に該当しないかを確認してみてください。
①Windowsなど、MSX以外のマシンでフォーマットしたフロッピーディスクをドライブに入れて起動した場合、MSX用ではないプログラムが実行されるために止まってしまいます。そのようなディスクは、BASICの画面が表示されてから挿入してください。
②1DDフォーマットのフロッピーディスク(「MSX-DOS TOOLS」など)は、Windows側の仕様により読み込むことはできません。
③プロテクトのかかった市販ソフトで、それが利用しているハードウェアで正常な判別ができない場合、起動時に停止することがあります。この場合の対処はできません。
④古いフロッピーディスクの場合、データが壊れて正常に動作しない場合があります。

Q WindowsXPで、2DD(720KB)ディスクのフォーマットができない！

WindowsXP以降では2DDは非対応となっている場合があります。また、読み込みはできてもフォーマットできない機種もあります。裏技ですが、Windowsのフォーマットメニューに2DD(720KB)がない場合でも、スタートボタンの「ファイル名を指定して実行」で「FORMAT A:」と入力することで2DDフォーマットができる場合もあります。

Q 2HD(1.44MB)のフロッピーディスクを使うことは可能？

どうしても2DD(720KB)のフロッピーディスクがない場合、2HD(1.44MB)のフロッピーディスクのライトプロテクトノッチとは反対側の穴を、セロハンテープなどでふさぐことで代用できます。この場合、穴をふさいだ後に2DDフォーマットしてから使用してください。

Q USB接続のフロッピーディスクドライブは使えるの？

使えます。ただしプロテクトの判別はできないことが多いようです。フロッピーディスクからPCがブートできるタイプのものであれば、判別できる可能性は高くなります。

Q フロッピーディスクの読み込みが異常に遅くなるのですが？

まれにフロッピードライブとフロッピーディスクとの相性により起こる場合があります。

>>ジョイスティック／マウス／キーボード／その他の周辺装置

Q MSX用のジョイスティックを接続する方法ってあるの？

Windows用としてそのような製品があれば可能です。現在、市販の製品としては確認できませんでした。

Q 外付けのペンタブレットは使えるの？

「マウス版」ではペンタブレットは使用できません。PCに接続したマウスを使用して操作を行ってください。なお、「ジョイパッド版」のソフトウェアキーボードに関しては、ペンタブレットで操作ができます。

Q キーボードの同時押し(カーソルキーの斜め方向など)が効かない！

Windowsでは、機種によって2つ以上のキーを同時に判別できない場合があります。そのため、ジョイスティックを使用するか、スキン上のジョイパッドにある斜め方向キーを使用してください。

Q マウス対応のプログラムは使えるの？

「マウス版」を起動することで使用できます。この場合、ポート1のみの対応となります。ポート2を必要とする場合は使用できません。

Q スキン上のジョイパッドが使えない！

ソフトがジョイパッドに対応していない可能性があります。また、「マウス版」の場合には、ジョイパッドが無効になっています。

Q Windows用のジョイスティックは接続して使えるの？

ジョイスティックはゲームポート用でもUSB接続用でも、どちらも使用可能です。また、2台まで(「ジョイパッド版」の場合)つなぐことができます。その場合、ポート1／2の区別は、「コントロールパネル」にある「ゲームコントローラ」の優先するデバイスを選択して設定してください。なお、MSXではトリガーボタンを2つしか使いません。

Q マウスが使えないんだけど！

「マウス版」を起動している状態で[F12]キーを押すと、WindowsのマウスがMSXPLAYer側のマウスとして使えるようになります(Windowsのマウスポインタは消える)。再度[F12]キーを押すことでWindowsのマウスポインタが動かせるようになります。

↑「マウス版」を起動した状態なら、[F12]キーを押すことで、MSXPLAYerのマウスとして使うことができる。

目指せクリエイター Part 2

初めて作るミニゲーム

MSX-BASIC講座

本書でMSXのゲームの世界を充分に堪能したら、ぜひともプログラミングに挑戦してほしい。MSXでプログラミングを学んだ人たちの多くが、現在ゲーム業界を始めとする、各種コンピュータ業界で活躍しており、そのクリエイター精神は現在においても引き継がれている。MSXPLAYerによって、再びクリエイター精神を見直してみようではないか。

TEXT/花岡 朋和

BASIC版
MSXPLAYerに収録

プログラムを入力してみよう

プログラミングを学ぶにあたって、1本のサンプルプログラムを用意した。ここではタイトルを「きらきら星」とする。内容は、ごくごくシンプルなシューティングゲームの原点ともいえるミニゲームである。

プログラムはMSXPLAYerのBASIC版ディスクAにも収録されているので、そちらを実行してもらえばよいのだが、プログラムを組むにあたっては、目だけでなく体を使って覚えるのもよい方法だ。要するに、キーボードを使ってプログラムを入力する作業を経験することが、おおいに役立つということだ。

MSXPLAYerのBASIC版を起動すると、画面1のような画面が表示される。これを確認すれば、あなたはBASIC言語への入り口に立ったことになる。落ち着いて、リスト1のプログラムをキーボードから打ち込んでほしい。その際、気をつけるべき点を紹介しておこう。

画面1　MSXPLAYer BASIC版

行番号

行の最初は、必ず数字（行番号）で始まる。BASICの大きな特徴の1つが「行番号」を採用していることだ。BASICでは、プログラムを実行する際には行番号の順番に従って実行していくのが基本となる。もちろん、途中の命令によって違う行番号へ処理を飛ばすこともできる。ちなみに、行番号と命令のあいだにはスペースを1つ以上入れる必要がある。

入力の締めは[リターン]キー

1行入力したあとには、必ず[リターン]（WindowsのEnter）キーを押す。そうして初めてその1行がコンピュータ内のメモリにプログラムとして記憶されるのである。ワープロで[Enter]キーを押すのとは意味が違うので、行の途中で[リターン]キーを押しても行は分割できないので注意。

間違えた場合の修正

間違えた場合の修正には、カーソルキー、[INS]キー、[DEL]キー、[BS]キーなどを使う。それぞれのキーは、Windowsの矢印キー、[Insert]キー、[Delete]キー、[Backspace]キーに相当する。これらのキーについては、ワープロソフトやテキストエディタを使ったことがあれば、すぐに理解できるだろう。

[INS]キーを押すと、いわゆる「挿入モード」になって、文字入力位置を示すカーソルの形状が変わる（半分の高さになる）。もう一度押せば、上書きモードに戻る。

[DEL]キーはカーソル位置の1文字を消去、[BS]キーはカーソルの左側の1文字を消去だ。

ちなみに、いったん[リターン]キーを押してしまった行については、カーソルキーを使ってカーソルを移動させ、修正したあとで改めて[リターン]キーを押して記憶させればよい。

大文字小文字

MSX-BASICでは、アルファベットの大文字・小文字は区別されない。BASICは基本的に大文字で記述するのが慣わしで、たとえ小文字で入力しても、メモリの中では自動的に大文字へと置き換えられる。もちろん例外もあり、たとえば引用符でくくられた部分については大文字・小文字が区別される。ただし、このプログラムにおいてはどちらでも問題ないので気にする必要はないだろう。

プログラムを実行してみよう

プログラムのセーブとロード

さて、時間がかかったとは思うが、すべて入力し終わっただろうか？ ならばすぐに実行……、といきたいところだが、その前にやっておかなければならない大事なことがある。それは、プログラムをセーブ（保存）しておくことである。

ここで、セーブの説明をする前に、MSXPLAYerでやってほしいことがある。それは、MSXPLAYerウィンドウの下に並ぶ、ディスクアイコンのCをクリックするというもの。これで、保存先が仮想ディスクCに変更される。なぜ、このようなことをするかと言うと、仮想ディスクAには本書が用意したサンプルプログラムが収録されているため、それを上書きしてしまうことになるからだ。ちなみに、仮想ディスクAはBASICプログラム満載、仮想ディスクBはMSX-DOS起動用で、仮想ディスクCとDは空きディスクだ。

さて、話をセーブに戻そう。セーブするには以下の命令を入力して［リターン］キーを押す。

SAVE "KIRAKIRA.BAS"

引用符でくくられた部分は保存先のファイル名を示している。ここではKIRAKIRA.BASとしたが、もちろん各自適当な名前をつけてよい。ただし、ファイル名は英数字8文字以内という規則がある。ピリオド以降の部分を「拡張子」と呼ぶ。拡張子に使えるのは、英数字3文字以内である。ここでは、BASICプログラムであることがすぐ判別できるように「.BAS」という拡張子をつけてみた。

ちなみに、ディスクからプログラムを読み出すには以下の命令を入力すればよい。

LOAD "KIRAKIRA.BAS"

引数については、SAVEと同様なので説明の必要はないだろう。

また、ファイル名がわからないときは以下の命令を入力する。

FILES

これは、ディスクに記録されているファイルの名前を表示する命令である。

プログラムの実行とエラー

少々話はそれたが、セーブが完了したらいよいよプログラムを実行することにしよう。プログラムを実行させる命令は、

RUN

である。現在でも「プログラムが走る」という言い方をするので、覚えやすいはずだ。

しかし、一発でプログラムがきちんと動くことはまれである（動いた人はおめでとう）。たいていは「ピッ」という音（ビープ音）とともに、何かしらのメッセージが出てプログラムは停止してしまう。このメッセージを「エラーメッセージ」と言う。たとえば、

Syntax error in 100

と表示されたとしよう。「Syntax error」とは「構文の誤り」を示すエラーメッセージなのだが、要するにどこかでタイプミスをしているよ、ということである。「in」のあとの数字は行番号を示している。この場合は、100行のどこかにタイプミスが含まれていると考えればよい。ならば、さっそく100行を呼び出して修正しよう。

LIST 100

この命令を実行すれば、100行の内容が画面に表示される。タイ

表1 LISTコマンドの指定と表示

LIST	すべての行番号を表示
LIST -100	先頭から100行までを表示
LIST 100-200	100行から200行までを表示
LIST 200-	200行から最後までを表示

リスト1

```
100 SCREEN 5,2
110 BG$="CCGGAAGRFFEEDDCRGGFFEEDRGGFFEEDRCCGGAAGRFFEEDDCR"
120 FOR J=0 TO 2
130 C$="":D$=""
140   FOR I=1 TO 16
150     READ A$,B$
160     C$=C$+CHR$(VAL("&B"+A$))
170     D$=D$+CHR$(VAL("&B"+B$))
180   NEXT
190 SPRITE$(J)=C$+D$
200 NEXT
210 TF=0:TX=0:TY=0
220 X=120:SC=0
230 A=RND(-TIME)
240 CT=0
250 PLAY"T128L8"
260 GOSUB 500
270 ON INTERVAL=30 GOSUB 500
280 INTERVAL ON
290 ON STRIG GOSUB 570
300 STRIG(0) ON
310 IF CT=48 THEN SCREEN 1:PRINT "SCORE=";SC:END
320 IF RF=1 THEN PUT SPRITE 0,(RX,RY),10
330 PUT SPRITE 1,(X,194),9
340 IF TF=0 THEN 440
350 PUT SPRITE 2,(TX,TY),15
360 IF ABS(RX-TX)<8 AND ABS(RY-TY)<8 AND RF=1 THEN 370
ELSE 420
370 PLAY "","O6C8"
380 SC=SC+10
390 TF=0:RF=0
400 PUT SPRITE 0,(0,217),10
410 PUT SPRITE 2,(0,217),15
420 TY=TY-15
430 IF TY<0 THEN TF=0:PUT SPRITE 2,(0,217),15
440 ST=STICK(0)
450 IF ST=7 THEN X=X-8
460 IF ST=3 THEN X=X+8
470 IF X<0 THEN X=0
480 IF X>240 THEN X=240
490 GOTO 310
500 CT=CT+1
510 PLAY MID$(BG$,CT,1)
520 IF CT MOD 4<>1 THEN 560
530 RF=1
540 RX=INT(RND(1)*240)
550 RY=INT(RND(1)*178)
560 RETURN
570 IF TF=0 THEN TF=1:TX=X:TY=184
580 RETURN
590 DATA 00000001,00000000
600 DATA 00000001,00000000
610 DATA 00000011,10000000
620 DATA 00000011,10000000
630 DATA 00000111,11000000
640 DATA 00000111,11000000
650 DATA 11111111,11111110
660 DATA 01111111,11111100
670 DATA 00011111,11110000
680 DATA 00001111,11100000
690 DATA 00001111,11100000
700 DATA 00011111,11110000
710 DATA 00011110,11110000
720 DATA 00111000,00111000
730 DATA 00100000,00001000
740 DATA 00000000,00000000
750 DATA 00000011,00000000
760 DATA 00001111,11000000
770 DATA 00011111,11100000
780 DATA 00111111,11110000
790 DATA 00111111,11110000
800 DATA 01111111,11111000
810 DATA 01111111,11111000
820 DATA 01111111,11111000
830 DATA 01111111,11111000
840 DATA 11111111,11111100
850 DATA 11111111,11111100
860 DATA 11111111,11111100
870 DATA 11111000,01111100
880 DATA 11100000,00011100
890 DATA 11000000,00001100
900 DATA 10000000,00000100
910 DATA 00000011,00000000
920 DATA 00000011,00000000
930 DATA 00000011,00000000
940 DATA 00000011,00000000
950 DATA 00000011,00000000
960 DATA 00000011,00000000
970 DATA 00000011,00000000
980 DATA 00000011,00000000
990 DATA 00000011,00000000
1000 DATA 00000011,00000000
1010 DATA 00000000,00000000
1020 DATA 00000000,00000000
1030 DATA 00000000,00000000
1040 DATA 00000000,00000000
1050 DATA 00000000,00000000
1060 DATA 00000000,00000000
```

プミスを見つけたら、カーソルキーを使って修正して、[リターン]キーで記憶させよう。これで修正完了である。ちなみに、[F4]キーには“list”があらかじめ登録されているので、[F4]を押して、行番号のみ指定すれば手間が省ける。

LIST命令には、他にもいろいろな指定の方法があるので**表1**にまとめておく。

その他のエラーメッセージについても、代表的なものをいくつかあげておこう。

Illegal function call
NEXT without FOR
Out of DATA
Type mismatch
Undefined line number

詳細は本書の巻末資料に「エラーメッセージ表」が掲載されているので、そちらを参照していただきたい。基本的には該当する行番号の中にミスが潜んでいる場合が多いが、まれにそうではないことがある。上にあげた中では「NEXT without FOR」と「Out of DATA」にその可能性が高い。これらのエラーメッセージはFORとNEXT、READとDATAという、必ず対になって使われる2つの命令の関係がうまくいっていないことを示している。なので、該当する行だけでなく、対となるもう一方の命令がある行も調べなければならないというわけだ。気をつけていただきたい。

バグとデバッグ

エラーメッセージが出なくなれば、無事終了…といきたいところだが、必ずしもそうはならない。エラーとなって表面上に現れないミス、すなわち「バグ」が潜んでいることがあるからだ。これについては、プログラムの流れを理解して、地道につぶしていくしかない。バグをつぶす作業のことを「デバッグ」と呼ぶが、実際のソフトウェア開発に際してもデバッグは非常に手間のかかる仕事である。おそらく、市場に出回っている市販のソフトウェアの中でもバグが1つもないものは皆無といってよいだろう。コンピュータとつきあうということは、すなわちバグとつきあうことでもあるのだ。

もちろん、このプログラムはリストも短いので、バグをすべてつぶすことは難しくない(そもそもバグはつぶしてある)。リストと画面をよ〜く見比べれば、あなたにも必ずデバッグできるはずだ。Good Luck！

「きらきら星」の遊び方

ここで、サンプルプログラムKIRAKIRA.BASの遊び方を紹介しておこう。

実行すると、まず、画面下に自機が表示される。カーソルキーの左右で自機が移動し、[スペース]キーで弾を発射することができる。

画面上のほうに表示されている星を撃つと10点の得点が与えられるが、星はおよそ2秒経過すると、他の位置に移動してしまう。

BGMの「きらきら星」の演奏が終了するとゲームオーバー。得点は最後に表示される。

MSXPLAYerには実行速度を調節する機能があるので、速い「Fast」モードでもぜひ実行してみてほしい。BASICの処理速度の遅さを補ってくれ、まったく違ったゲームが楽しめるはずである。

プログラムKIRAKIRA.BASの仕組み

さて、いよいよこのプログラム「きらきら星」がどのような処理の組み合わせによって動作しているかを解説する。個々の命令の詳細については、誌面の関係上ここには載せきれないので、巻末資料「MSX-BASICリファレンス」を参照していただきたい。

表2 変数の型

型	内容	末尾の記号
整数型	−32768〜+32767の範囲内の整数	%
単精度実数型	小数点以下も含めて6ケタ以内の実数	!
倍精度実数型	小数点以下も含めて14ケタ以内の実数	#
文字型	0〜255文字の文字列	$

表3 変数

A$, B$, C$, D$	スプライトパターン定義用
BG$	BGMデータ
A	乱数初期化用（ダミー）
CT	ゲームの経過時間＆BGMの演奏箇所の指定
I, J	ループカウンタ
RF	星が画面内に存在しているかどうかのフラグ
RX, RY	星のX座標，Y座標
SC	得点（SCOREの略）
ST	カーソルキーの入力受付用
TF	弾が発射されているかどうかのフラグ
TX, TY	弾のX座標，Y座標
X	自機X座標

変数 —プログラムを覗く前に

変数とは、プログラム中で使用するデータを格納する「箱」のようなものと考えていただければ分かりやすいだろう。変数の中に格納されているデータを計算したり、加工することによってさまざまな処理が進んでいく。

変数を使うには「名前」と「データの型」を決めなければならない。名前は、

1.アルファベット1文字（例：A）
2.アルファベット2文字（例：AB）
3.アルファベット1文字＋数字1文字（例：A1）

のいずれかの形式をとる。3文字以上の名前を指定した場合には、3文字目以降は無視される。また、カタカナ・ひらがなは変数名に使うことができない。アルファベットの大文字・小文字は区別されない。

データの型は、変数名のあとに記号1文字をつけて表す。MSX-BASICで定義されている型は、表2に示す4種類がある。

表2の上3つの型はいずれも数値を扱う型である。なぜ3つに分かれているかというと、それぞれ使用するメモリの量が違うからだ。それぞれの関係は、整数型＜単精度実数型＜倍精度実数型となる。

当然のことながら、使用するメモリが少ないほど処理速度は速くなるので、小数点以下の計算が必要なければ整数型の変数を使うことによって速度が向上する。

最後の文字型は、数値を扱う上記3つの型とはまったく別世界だと考えたほうがいい。文字型同士は足し算することはできるが、引き算・掛け算・割り算はできない。その代わり、文字列を扱うための専用の命令が多数用意されている。

なお、記号を省略したときの初期設定（デフォルト）は倍精度実数型となっている。

また、同じ変数名でも、型が違えば別の変数とみなされる。例えば、変数A%、A!、A#、A$はすべて別の変数なので、同じプログラム内で共存することができる。

以上のお約束事を覚えたら、あとはプログラムリスト中で、どのように使われているかを実際に見てみよう。

このプログラムで使われている変数を**表3**にまとめておく。一見しただけでは意味が分からないかもしれないが、順次補足していきたい。

スクリーンモードの初期化

100〜110行では、プログラムの初期設定を行っている。

100行のSCREEN文は、スクリーンモードを5、スプライトの大きさを16×16ドットに設定するという意味である。MSXには、0〜12までのスクリーンモードがあり、それぞれ画面に表示できる文字やドット、色の数が異なっている。詳細は巻末資料参照のこと。「スプライト」については、次の項で説明する。

なお、110行は、このプログラムで使用するBGM「きらきら星」のデータを変数BG$に格納している。

スプライトパターン定義

120〜200行では、スプライトパターンの定義を行っている。

```
120 FOR J=0 TO 2
130 C$="":D$=""
140   FOR I=1 TO 16
150     READ A$,B$
160     C$=C$+CHR$(VAL("&
B"+A$))
170     D$=D$+CHR$(VAL("&
B"+B$))
180   NEXT
190 SPRITE$(J)=C$+D$
200 NEXT
```

スプライトとは、キャラクタを動かす際に便利な機能である。1個のスプライトは、8×8ドットあるいは16×16ドットのパターンで構成されている(このプログラムでは後者を指定している→100行)。MSXでは、1画面に同時に32個のスプライトを表示することが可能で、それぞれ0〜31の番号(スプライト面番号)が与えられている。

スプライト面番号は、同時に優先順位を示している。例えば、面番号0のスプライトと面番号1のスプライトが重なった場合には、面番号0のスプライトが手前に表示され、面番号1のスプライトは後ろに隠れることになる。

SPRITE$を使ったパターンの設定については、巻末のリファレンスを参照してもらうとして、ここではプログラミング全般において大切なループ(繰り返し)を使った、パターンデータの読み込みについて説明しておきたい。

ループ

ループを形成するのは、FOR文とNEXT文という2つの命令の組み合わせである。この箇所では、まず120行のFOR文と200行のNEXT文の組み合わせに着目する。これは、120行〜200行のあいだの処理を一定回数繰り返せ、という意味である。何回繰り返すかは120行のFOR文に書かれている。「FOR J=0 TO 2」は、以下の意味を表す。

1.まず、変数Jに0を入れて、1回目の処理を行う。
2.処理が終了したら、変数Jの値に1を加えて次の処理に入る。
3.2の作業を繰り返し、変数Jの値が2の時点で終了。

このループは変数Jの値が0、1、2のとき――すなわち3回、同じ処理を繰り返すことになる。もちろん変数Jの値は他の命令で取り出すことができる。ここでは190行のSPRITE$文が使用している。

また、このループの中には、140行のFOR文と180行のNEXT文というもう1つの組み合わせが存在しており、すなわち「2重ループ」を形成していることも重要なポイントである。この場合、内側のループは外側のループの中に完全に収まっていなければならない(入れ子構造)。

基本的なループの仕組みが理解できただろうか。なお、FOR〜NEXT文以外にもループを作る方法はあるので、後ほど説明したい。

FOR〜NEXT文以外にも、対で使用する命令が存在する。150行のREAD文と、590行以降のDATA文である。DATA文は、文字通り「データ」を置いておくための命令で、READ文はそのデータを「読み出す」命令である。150行のREAD文を実行するたびに、最初のほうから順次データが変数へと読み込まれていく。ここではA$、B$と2個の変数を指定しているので、データも2個ずつ読み込まれることになる。

変数の初期設定

210〜250行はゲームに必要な変数の値を改めて設定している部分である。

```
210 TF=0:TX=0:TY=0
220 X=120:SC=0
230 A=RND(-TIME)
240 CT=0
250 PLAY"T128L8"
```

C言語などでは、あらかじめ使用する変数を「宣言」しておく必要があるのだが、BASICではその必要はない。数値型の変数の場合、初期値は0となるので、210行のようにわざわざ0を入れることは実はほとんど意味がないのだが、一種の気休めのようなものと思ってほしい。

230行は「乱数」の初期化を行っている箇所である。乱数についてはあとで説明する。

250行では効果音の初期設定だ。詳細は巻末資料のPLAY文を参照してほしい。

サブルーチン

260行のGOSUB文は「500行に飛べ」という意味の命令である。GOSUB文は、必ずRETURN文と対になって使われ、RETURN文が実行されると処理はもとに戻る。

このプログラムでは560行にRETURN文が存在する。この場合、500〜560行の範囲を「サブルーチン」と呼ぶ。サブルーチンとは、普段実行されているプログラム(メインルーチン)に対して、必要な際に呼び出されて実行される別のプログラムといった意味である。何度も同じ処理を繰り返す必要がある場合、その処理をサブルーチンにしておくと便利であるだけでなく、プログラムそのものの短縮にもつながる。

割り込み

270〜300行では、割り込み定義を行っている。

```
270 ON INTERVAL=30 GOSU
B 500
280 INTERVAL ON
290 ON STRIG GOSUB 570
300 STRIG(0) ON
```

本来、プログラミングの基本を学ぶうえでは割り込みという機能の紹介は後回しにするほうが賢明かもしれない。しかし、コンピュータそのものの仕組みを学ぶうえで「割り込み」の概念を知っておくことは非常に重要であり、あえて取り上げておく。

Windowsの登場はパソコンの概念を大きく変えたが、その1つにWindowsがマルチタスクに対応したOSだということがあげられる。マルチタスクとは、同時に複数のプログラムが動作することだ。我々は、ふだん何も考えることなく、当たり前のように同時に複数のウィンドウを開き、使用している。

しかし、厳密にはこの定義は正しいとはいえない部分もある。というのは、実際にはコンピュータは同時に複数のプログラムを実行することはできないのだ(もちろん例外も多数あるので技術者の方はツッコミ入れないように)。マルチタスクを実現するためには、実は適当なタイミングで順次実行するプログラムを切り換える必要がある。それにより、あたかも同時に実行しているように見せかけているわけである。

この、プログラムを切り換えるために必要な「適当なタイミング」を指示するのが「割り込み」という機能だと考えればいい。割り込みにはいろいろな種類があるので、ここでは詳しく説明はしないが、プログラミングに興味のある方ならば押さえておいて損はない。

このプログラムで使用している割り込みは以下の2種類である。

タイマ割り込み

タイマ割り込みには、ON INTERVAL文、INTERVAL ON文を使う。

タイマ割り込みは、一定時間が経過すると自動的に割り込みが発生する。MSX-BASICでは1/60秒単位で指定する。270行では30/60秒＝1/2秒ごとに割り込みが入るように指定している。280行のINTERVAL ON文が実行された時点から、時間のカウントが開始される。

割り込みが入ると、その時点でどんな処理をしているかに関わらず、強制的に500行へと処理が飛ぶ。GOSUBがついていることからもわかるように、割り込み前の処理と割り込み後の処理はメインルーチンとサブルーチンの関係になっている。RETURN文が実行されることによって、割り込み前の行に処理が戻る。

トリガ割り込み

トリガ割り込みには、ON STRIG文、STRIG ON文を使う。

トリガ割り込みは、[スペース]キー、またはジョイスティックのトリガボタンが押されたのを感知して割り込みが発生する。300行のSTRIG ON文が実行された時点から検出が始まる。(0)は[スペース]キーを意味している。つまり、[スペース]キーが押された場合のみ割り込みの対象となる。

プログラムの終了

310行では、ゲームオーバーかどうかの判定とゲームオーバーの処理を行っている。

プログラム全体をよく見ると、310行～490行が大きなループになっていることがわかるはず。変数CTは、BGMのどの時点を演奏しているかを示すカウンタとして用いられている。その値が48になったとき──つまり、BGMが終了したときがゲームオーバーのときである。ゲームオーバーと判定されれば、スクリーンモードを1に変更し、得点を表示したうえでこのプログラムは終了する。プログラムの終了にはEND文を使う。

スプライトの表示と座標

320行では、星の表示を行っている。

```
320 IF RF=1 THEN PUT SPRITE 0,(RX,RY),10
```

プレイヤーの標的となる「星」を表示する。スプライトを表示するためにはPUT SPRITE文を用いる。星にはスプライト面番号0を割り当てている。

なお、330行では、自機をスプライト面番号1を割り当てて表示し、350行では、弾をスプライト面番号2を割り当てて、それぞれ表示している。そちらも参照してほしい。

PUT SPRITE文でスプライトを表示するには、引数に（X, Y）の形式で表示する座標を指定する。画面左上が（0,0）になることに注意しよう。座標は、スクリーンモードによっても異なるが、スクリーンモード5の場合は図1のようになっている。

フラグ

星を表示するのは、変数RFの値が1のときに限られている。これは何を意味するのだろうか。変数RFは、画面上に星が存在するかどうかを数値で示すために使われている。値が1ならばON、0ならばOFFである。最初、星が表示されるときには当然値が1になっているが、星が撃ち落とされてしまった場合には次の星が表示されるまで値が0となる。

このような変数の使い方を「フラグ」と呼んでいる。ロールプレイングゲームが好きな方ならば、ゲーム用語としても定着しているので理解しやすいだろう。

条件分岐 ─当たり判定

340行では、弾が発射されているかどうかを判定し、プログラムを分岐させている。

```
340 IF TF=0 THEN 440
```

変数TFは、画面上に弾が表示されているかどうかを示すフラグである。値が0ならば表示されていないので350行以降の弾を移動させる処理は必要ない。そこで、一気に440行まで処理を飛ばしてしまう。

360行では、弾は星に命中しているかどうかを判定し、プログラムを分岐させている。

```
360 IF ABS(RX-TX)<8 AND ABS(RY-TY)<8 AND RF=1 THEN 370 ELSE 420
```

アクション系のゲームを作ろうとするならば避けて通れないのが「当たり判定」の処理。すなわち、あるキャラクターが他のキャラクターや障害物などと衝突しているかどうかを判定する処理である。この処理がうまくできるかによって、ときにゲーム全体の完成度が左右されることもあるので奥が深い。シューティングゲームが好きな人ならば、当たり判定が甘い（または辛い）という表現を一度は使ったことがあるだろう。

ここでは、弾が星に命中しているかどうかを判定している。当たり判定の基本は、お互いの座標の差をとることである。ABS関数は「絶対値」を返す関数なので、絶対値が8より小さいということは、両者の座標の差が－7～＋7の範囲に収まっていることを意味する。X座標、Y座標ともにこの範囲に収まっていれば命中となる。

ただし、忘れてはならないことがもう1つ。そもそも画面に星がなければ、絶対に命中することはない。3つ目の条件RF＝1が意味するのはこのことである。

命中していれば370行へ（実質次の処理に移るだけだが）、そうでなければ420行へ飛ぶ。

スプライトを消す ─命中処理

370～410行では、弾が命中した場合の処理を行っている。

```
370 PLAY "","O6C8"
380 SC=SC+10
390 TF=0:RF=0
400 PUT SPRITE 0,(0,217),10
410 PUT SPRITE 2,(0,217),15
```

370行で命中時の効果音を鳴らし、380行で得点を示す変数SC

図1：スクリーン座標

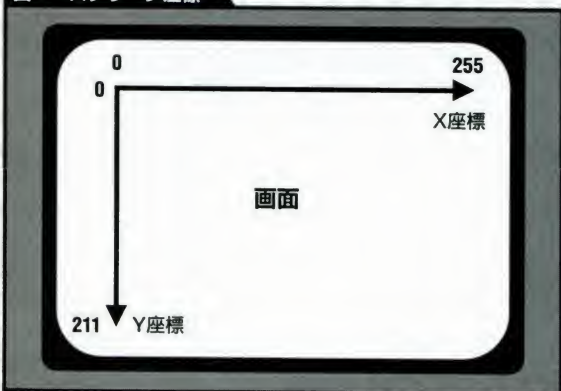
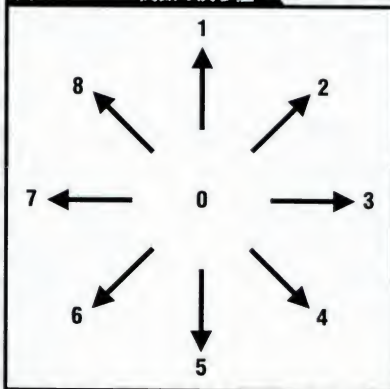



図2 STICK関数の戻り値

に10を加えている。390行では、弾・星双方のフラグをOFFにしている。実際にスプライトを消しているのは400〜410行。Y座標に217を指定しているが、単に画面の外に追い出しているのではなく、217は「スプライトを消せ」という意味を持つ特殊な値なのである。

移動スピード―弾の移動処理

420〜430行では、弾の移動処理を行っている（実際に表示するのは350行）。変数TYの値から15を引くことによって、上のほうに移動することになる。この数字を変えることによって、弾のスピードを変えることも可能だが、15を超えてしまうと当たり判定に支障をきたすのでこれ以上は速くできない。

変数TYの値がマイナスになるということは、画面から消えることを意味する。消えた場合にはフラグをOFFにして、スプライトを消している。

STICK関数―自機の移動処理

440〜490行では、STICK関数の戻り値に応じた自機の移動処理を行っている。

```
440 ST=STICK(0)
450 IF ST=7 THEN X=X-8
460 IF ST=3 THEN X=X+8
470 IF X<0 THEN X=0
480 IF X>240 THEN X=240
490 GOTO 310
```

440行でカーソルキーの入力を受付けている。返ってくる値については図2を参照のこと。その値によって、自機のX座標を示す変数Xを変化させる。画面の端に達した場合にはそれ以上動かないような処理になっているが、画面の反対側にワープさせるようにすればまた違う面白さを持ったゲームになるだろう。簡単にできるので、どうやればいいのかは各自で考えていただきたい。

サブルーチン1―BGMと星

500〜560行は、BGMの演奏と星の出現を処理するサブルーチンになっている。

```
500 CT=CT+1
510 PLAY MID$(BG$,CT,1)
520 IF CT MOD 4<>1 THEN 560
530 RF=1
540 RX=INT(RND(1)*240)
550 RY=INT(RND(1)*178)
560 RETURN
```

このサブルーチンは2か所から呼び出されていることにお気づきだろうか。260行のGOSUB文で最初の1回、それ以後はタイマ割り込みによって1/2秒ごとに呼び出される。

500行で、変数CTの値に1を加えている。この変数はカウンタとして使われていることは先に述べた。510行では、この変数CTの値をもとにBGMを演奏している。MID$関数は、文字列の途中の部分を取り出す関数である。詳細は巻末資料参照のこと。

520行で行っている条件分岐は一見難しく見えるが、よく使うので押さえておきたい。x回につき1回の割合である処理を行いたいという場合にこのテクニックは使われる。「CT MOD 4」とは、変数CTの値を4で割ったときの「余り」を意味している。その余りが1以外であれば、処理は560行に飛ぶ（RETURN文があるのでそのままサブルーチン終了）。すなわち、530〜550行の処理が行われるのは4回につき1回である。もともと1/2秒ごとに呼び出されるサブルーチンなので、1/2秒×4＝2秒に1回実行されると言い換えてもよいだろう。

乱数

530〜550行では、星を出現させる処理を行っている（実際に画面に表示するのは320行)。530行でフラグをONにし、540〜550行で座標を設定している。その際に使われているのが「乱数」の考え方である。乱数とは、文字通り「でたらめ（ランダム）な数」のこと。いわば、コンピュータの中でサイコロを振るようなものである。とくにゲームにおいては、偶然性がゲームの面白さを左右することが多いため、乱数は欠かせない存在である。

RND関数は「0以上1未満の乱数」を発生させる関数である。後ろの(1)には深い意味はない。INT関数は値の小数点以下を切り捨てる関数で、RND関数とは親友といってよいほどセットで使われる。540行で変数RXに入る値は0〜239の整数、550行で変数RYに入る値は0〜177の整数となる。これを応用することで、さまざまな範囲の乱数を作ることができる。例えば、オーソドックスな6面体のサイコロならば、INT(RND(1)*6)+1という表記で簡単に作れる。

ちなみに、230行では乱数の「初期化」をしていると書いたが、この行を削除するとどうなるか。答えをバラしてしまうと、何度ゲームをやっても同じパターンで星が現れてしまうのだ。これでは乱数の意味をなさない。しかし、これはある意味コンピュータが持つ限界をも示しており、なんだか微笑ましくもある。

サブルーチン2―弾の発射

570〜580行は、弾の発射を処理するサブルーチンになっている。

```
570 IF TF=0 THEN TF=1:TX=X:TY=184
580 RETURN
```

[スペース]キーが押されたときに呼び出される（トリガ割り込み)。画面内にすでに弾が存在していない場合に限り、フラグをONにし、座標を設定している。実際に弾を画面に表示するのは350行である。

スプライトパターンの定義

590〜1060行は、DATA文によるスプライトパターンデータの定義になっている。

590〜740行が星、750〜900行が自機、910〜1060行が弾のデータである。すべて0と1のみの、いわゆる「2進法」で表記されている。このデータをいじることによって、それぞれのキャラクタの形を自由に変えられるので、デザインセンスのある方はぜひお好みのキャラクタに変えて遊んでいただきたい。

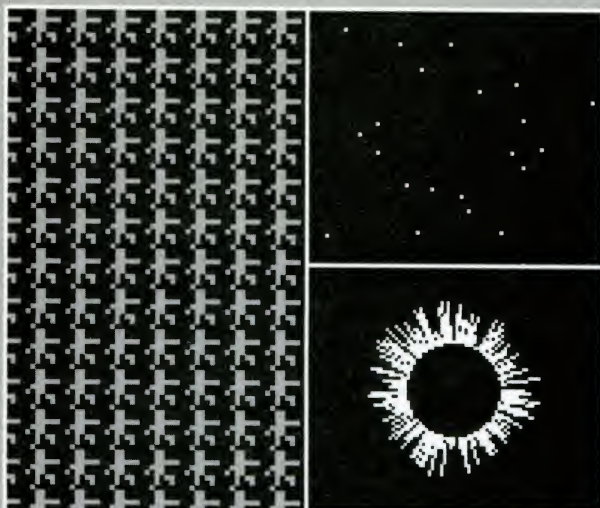
★

以上、ページ数の都合もあり駆け足の解説になってしまったが、基本的な処理は理解していただけたのではないかと思う。

終わりに

これで、ひととおりのプログラムの解説を終了する。ページ数の都合もあり、あまり詳しく触れることはできなかったが、どんなに複雑なプログラムでも、1つ1つの小さな処理の集まりでできあがっていることを実感していただけたかと思う。しかし、このプログラムはゲームとして見たときには未熟な点が多々ある。そこをどう面白くするかは皆さんの腕の見せどころだ。

このコーナーを読んでくれたすべての方にクリエイターになることを勧めるつもりはないが、この記事が自分とコンピュータとのつきあい方を考え直すきっかけになってもらえれば幸いである。

10年後、20年後のコンピュータ業界を支える人物が輩出されることを祈りつつ、この稿を終わりとしたい。今度は次号の「中級編」でお会いしましょう（次号があればの話であり、あくまでも予定）。

短いけれどくふう満載
ショートプログラム

MSXはひと昔前のパソコンなだけに、当然現在のものに比べるとはるかに表現力が貧弱である。しかし、使用するデータ量も少ないだけに、軽い気分でプログラミングを楽しむことができる。このコーナーでは、ごく短いプログラムで「くすっ」と笑っていただけるような小さな作品を3本紹介する。この意図が少しでも伝わり、読者の皆さんに「わびさび」にも似た感覚を味わっていただければ幸いである。

なお、このコーナーのプログラムは、いずれもMSXPALYerのBASIC版仮想ディスクAに収録されている。

TEXT/花岡 朋和

兵隊の行進 SP1.BAS

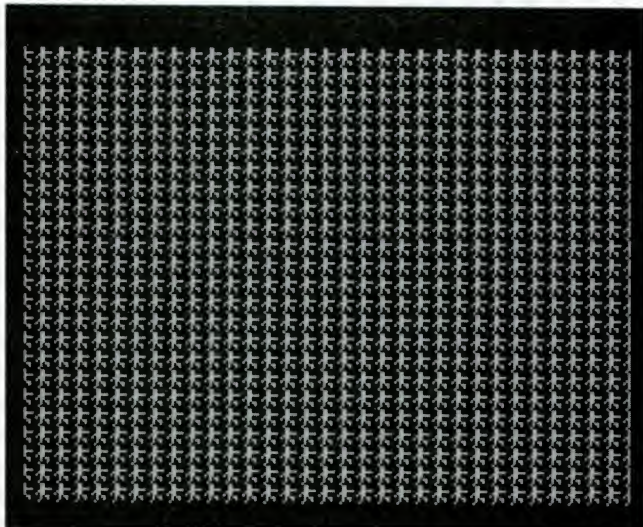
このプログラムを実行すると、画面いっぱいに並んだ人間たちが左→右へと行進を始める。そうは見えない？いや、そうだと思ってほしい。

画面全体に表示されている人間は32×24＝768人。このプログラムでは、その768人を1人ずつ動かしているのかというと、そうではない。実際には、1人たりとも動かしていないというのが正解だ。

どのように処理しているのかというと、画面全体に敷き詰められているのは「@」というただ1種類の文字である。しかし、MSXではその文字のパターンを自由に書き換えることができる。そのことを利用し、連続してパターンを書き換えることによって、あたかも全体が動いているかのように見せるわけだ。ここでは横方向のみの動きだが、パターンを工夫することによって、縦やななめに動かすこともできる。

このプログラムには終わりがないため、[CTRL]＋[STOP]キーで強制終了しよう。終了後に[@]キーを押してみると秘密がわかる。

リスト SP1.BAS

```
100 SCREEN 1:WIDTH 32:KEYOFF:COLOR 15,1,1
110 DIM A(2,7,7)
120 FOR I=0 TO 2:FOR J=0 TO 7
130 READ A$:A(I,0,J)=VAL("&H"+A$)
140 NEXT J,I
150 FOR I=0 TO 2:FOR J=1 TO 7:FOR K=0 TO 7
160 A=A(I,J-1,K):B=(A MOD 2)*128+A¥2:A(I,J,K)=B
170 NEXT K,J,I
180 A=0:B=0:GOSUB 300
190 VPOKE &H2000+8,&HC0:COLOR=(12,1,5,1)
200 FOR I=1 TO 23:PRINT STRING$(32,"@");:NEXT
210 FOR I=0 TO 31:VPOKE &H1800+23*32+I,64:NEXT
220 FOR I=0 TO 3
230 A=I:IF A=3 THEN A=1
240 GOSUB 300
250 IF A=0 THEN Z=100 ELSE Z=50
260 FOR T=1 TO Z:NEXT
270 B=B+1:IF B=8 THEN B=0
280 NEXT
290 GOTO 220
300 FOR J=0 TO 7:VPOKE 64*8+J,A(A,B,J):NEXT
310 RETURN
320 DATA 18,18,1F,38,58,16,12,20
330 DATA 18,18,1C,3A,38,14,12,12
340 DATA 18,18,1C,1C,18,18,14,18
```

変形宇宙飛行 SP2.BAS

このプログラムでは、Windowsのスクリーンセーバ「宇宙飛行」の再現を目標にした。描画が終了すると「ピー」という音がするので、[スペース]キーを押す。すると、突然画面がモノクロの世界へと変化し、宇宙旅行が始まる。

このプログラムでは、MSXの機能である「スクリーンモード5」を使用している。スクリーンモード5において使える色は16色だが、それぞれ計512色の中から任意の色を割り当てることができる。そこで、すべての色をいったん「黒」に指定し、順番に色を切り替えることによって動きを出そうというわけである。このテクニックは一般的に「パレットアニメーション」と呼ばれている。たいへん応用範囲が広いテクニックなので、例えば画面をフェードイン/フェードアウトすることなども可能だ。

終了するには、[CTRL]+[STOP]キーで強制終了する。

準備段階では各色で渦巻き状の模様を描いておき、パレットを切り替えると宇宙旅行に見える。

リスト SP2.BAS

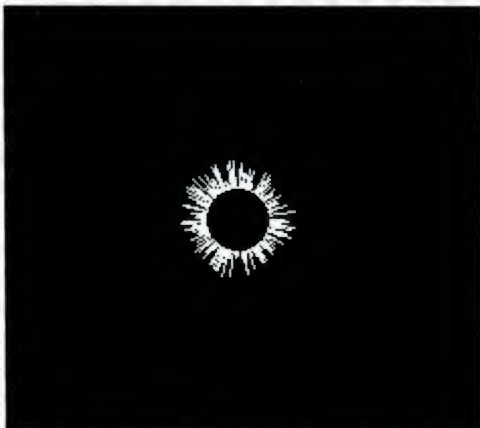
```
100 SCREEN 5:SETPAGE 0,0:COLOR 15,1,1:CLS
110 DEF FNA(A)=3.14159/180*A
120 R=RND(-TIME)
130 FOR I=0 TO 350 STEP 10
140 X=128:Y=106:A=I:S=15
150 C=INT(RND(1)*12)+2
160 GOSUB 310
170 S=3
180 A=A+6:IF A>=360 THEN A=A-360
190 S=S+0.6
200 GOSUB 310
210 F=(X<0)+(X>255)+(Y<0)+(Y>211)
220 IF F=0 THEN 180
230 NEXT:PLAY"V1505C8"
240 IF STRIG(0)=0 THEN 240
250 FOR I=2 TO 13
260 COLOR=(I,7,7,7)
270 FOR J=1 TO 20:NEXT
280 COLOR=(I,0,0,0)
290 NEXT
300 GOTO 250
310 X=X+COS(FNA(A))*S:Y=Y+SIN(FNA(A))*S
320 C=C+1:IF C=14 THEN C=2
330 PSET(X,Y),C
340 RETURN
```

皆既日食 SP3.BAS

天体ショーのひとつである「皆既日食」の雰囲気を味わうプログラムである。

このプログラムでも準備の過程を見せている。使用しているのはやはりスクリーンモード5だが、このモードでは画面を計4ページ分記憶しており、必要に応じて切り替えることができる。

まず、ページ番号0の画面に「コロナ」と呼ばれる部分と、その前後の丸い太陽を先に描画している。その後、表示画面はページ1に切り替わるのだが、実際の作業はページ0で行っている。太陽を別の場所にいったんコピーし、そこに黒丸(月の影)を重ねる。そうしてできあがった画面を、さらにページ1にコピーする。この作業の連続でアニメーションを表現している。わざわざこのような手間をかけるのは、MSXは描画速度が遅いため、作業過程を隠す必要があるからだ。

このプログラムは「ピー」という音が鳴ったあと、[スペース]キーを押すと終了する。

リスト SP3.BAS

```
100 SCREEN 5:COLOR 15,1,1
110 SETPAGE 0,0:CLS
120 A=RND(-TIME):P=3.14159
130 FOR I=0 TO 358 STEP 2
140 A=INT(RND(1)*15)+15
150 X=INT(COS(P/180*I)*A)
160 Y=INT(SIN(P/180*I)*A)
170 LINE(50,50)-STEP(X,Y),15
180 NEXT
190 CIRCLE(50,50),15,1
200 PAINT(50,50),1,1
210 CIRCLE(150,50),15,15
220 PAINT(150,50),15,15
230 SETPAGE 0,1:CLS:SETPAGE 1,0
240 COLOR=(2,0,0,0)
250 FOR I=20 TO 80
260 X=-(I<48 OR I>52)*100
270 COPY(X,0)-STEP(99,99) TO (0,100)
280 Y=INT((50-I)/2+150)
290 CIRCLE(I,Y),15,2
300 PAINT(I,Y),2,2
310 A=ABS(50-I):C=7+(A=1)*2+(A=2)*6
320 COLOR=(15,C,C,C)
330 COPY(0,100)-(99,199),0 TO (78,56),1
340 FOR J=0 TO 99
350 NEXT J,I
360 PLAY"V1505C8"
370 IF STRIG(0)=0 THEN 370
```

MSXミュージシャン講座

MuSICAによる音楽制作

パソコンでは音楽を演奏させることができるのはみんなも知ってのとおり。MSXの現役時代からそれは変わらない。もちろん、MSXPLAYerでも音楽演奏は可能だ。BASICプログラムを打ち込んで、自分の思いどおりに演奏できたときは感涙ものだぞ！ さらに「MuSICA」を使えば、自作ゲームに組み込める音楽を作成することも可能だ！ MSXサウンドクリエイターに必要な能力は、やる気と根気のたったの2つだ。ぜひ挑戦してほしい！

BASIC版
MSXPLAYerに収録

MSXを使った演奏のキホン

音楽を演奏する人は楽譜を見て楽器を演奏するけれど、MSXPLAYerは、音楽専用のコマンドであるMML（ミュージック・マクロ・ランゲージ）を読んでMSXの楽器を演奏する。さすがにMSXPLAYerは楽譜を読めないので、MMLに変換するのはプログラマーであるボクらのお仕事。BASICで使えるMMLは巻末のBASICリファレンスの「PLAY」の項にあるので確認してほしい。このMMLを駆使して曲を作っていくのだ。ちなみにMMLでは大文字小文字は区別されない。見やすいように書いていこう。

なお、楽譜の読み方全部をこのページで書くのは容量的に無理なので、付属CD-ROMにPDFファイルで収録した「MSX Magazine 1992年夏号（抜粋）」の記事を参考にしてくれ！

さて、MSXの楽器のセットである音源は2種類ある。1つは「PSG音源」、もう1つは「FM音源」だ。PSG音源は最も基本的な音源でいちばん手軽に音を発生させることができる。FM音源は内蔵されている楽器や効果音のデータを利用してきれいな音を発生することができる。

PSG音源による演奏

まずはちょっとしたプログラムを書いてみよう。演奏をさせるにはBASICの「PLAY」という命令を使う。「PLAY "cde"」のようにPLAY命令のあとにMMLを直接書いてもいいけど、あとあと編集するときにたいへんになるから、MMLを文字型の変数に代入して演奏させよう。

「A・B・C・D・E・F・G」はそれぞれ音程を意味していて、それぞれ「ラ・シ・ド・レ・ミ・ファ・ソ」の音に対応している。"cde"とすると"ドレミ"と演奏されるわけだ。この音程を表す文字のあとには音の長さを書くことができる。この数を書かないと、次に説明する「L」コマンドで指定されたものとして演奏される。

「L」は音長、音の長さを指定するコマンド。"L1"は全音符、"L2"は2分音符というように、64分音符まで指定できる。なお、音程コマンドのあとに音の長さを書かなかったときには最初に指定した値が有効になる。たとえば"L8cdec4"と"c8d8e8c4"は同じように演奏されるわけだ。

「O」はオクターブを指定するコマンドだ。"O4"がいちばん中心のオクターブとなる。これは楽譜で言うとト音記号の下一線のドの音だ。"O5"にすると音は高くなり逆に"O3"とすると、音は低くなる。"O4L4cdefgabO5c"としてみるとドから順に高いドまで演奏される。

「R」は休符を意味するコマンド。この休符にはLコマンドの値は効かない。Rのあとに音長を指定しないと、必ず4分休符を指定したことになるので要注意だ。

「T」はテンポを指定するコマンドで、曲のテンポを変えることができる。"T120"で曲のテンポは四分音符=120のテンポとなる。

これだけのコマンドがあれば、とりあえず演奏はできる。サンプルとしてリスト1を打ち込んでみよう。これはPSG音源だけで演奏するプログラムだ。入力が終わったら、[F5]キーを押すか「RUN」と打ち込むと音楽が流れる。これ

図1 MMLと楽譜の対応

リスト1～4

●リスト1

```
10 'LIST 1 -ｶｴﾙ ﾉ ｳﾀ(PSG)-
100 DEFSTR A : RIN=0
110 A="S1 M10000"
120 A(0)="T120 L4 O5" : A(1)=A+A(0)
130 A(2)="cdef" : A(3)="edcr"
140 A(4)="efga" : A(5)="gfer"
150 A(6)="crcr" : A(7)="crcr"
160 A(8)="L8 ccddeeff" : A(9)="e4d4c4r"
1000 IF RIN=0 THEN 1010 ELSE 1100
1010 ' ﾀﾝｵﾝ
1020 FOR I=1 TO 9
1030   PLAY A(I)
1040 NEXT I
1050 END
1100 ' ﾘﾝｼｮｳ
1110 FOR I=1 TO 10
1120   PLAY A(I),"O4"+A(I-1)
1130 NEXT I
1140 END
```

●リスト2

```
10 'LIST 2 -ﾅﾐ ﾉ ｵﾄ-
100 SOUND 7,&B10110001
110 SOUND 6,10+RND(1)*4
120 PLAY "T50 S14 M20000 c1"
130 GOTO 110
```

●リスト3

```
10 'LIST 3 -ﾄﾞﾗﾑ ｻﾝﾌﾟﾙ-
100 SOUND 6,4 : SOUND 7,&B10110001
110 PLAY "T120 S1 L8"
120 FOR I=1 TO 3
130 PLAY "M500ccM2500cM500c"
140 NEXT I
150 PLAY "M500cM2500cc16c16c"
160 GOTO 120
```

●リスト4

```
10 'LIST 4 -ｶｴﾙ ﾉ ｳﾀ(FM)-
20 CALL MUSIC(0,0,1,1)
100 DEFSTR A : RIN=0
110 A="@6 V14
120 A(0)="T120 L4 O5" : A(1)=A+A(0)
130 A(2)="cdef" : A(3)="edcr"
140 A(4)="efga" : A(5)="gfer"
150 A(6)="crcr" : A(7)="crcr"
160 A(8)="L8 ccddeeff" : A(9)="e4d4c4r"
1000 IF RIN=0 THEN 1010 ELSE 1100
1010 ' TANON
1020 FOR I=1 TO 9
1030   PLAY#2, A(I)
1040 NEXT I
1050 END
1100 ' RINSYO
1110 FOR I=1 TO 10
1120   PLAY#2, A(I),"@5 O4"+A(I-1)
1130 NEXT I
1140 END
```

は図1のような感じで楽譜をMMLに変換した結果だ。

プログラムを簡単に解説すると、行160までMMLを定義して行1000以降でまとめて演奏をさせている。ここでは文字型配列変数のA(0)～A(9)にMMLを書き込んでいる。ちなみに、BASICのテクニックだけど、DEFSTR命令で文字型を指定する「$」を省略しているぞ。

ちゃんと演奏できたら、ちょっとずつプログラムをいじってみよう。たとえば、行120のTの値を変更してみよう。値を大きくすれば、テンポは速くなり、値を小さくすればテンポは遅くなる。

PSGはただ演奏させると、音量の変化がない音が演奏される。そのため、たとえば"CCCC"と演奏させてもドの音が4つ演奏されるわけでなく、長いドの音が演奏されるだけだ。1つの音に音量の変化をつけるのが「エンベロープ」だ。このエンベロープは「S」と「M」コマンドで設定できる。これはリスト1の行110で使用しているので、値を変えるとどのように変化するのか実感してみてくれ。値の内容はリファレンスのPLAYの項に書いてある。

おまけでちょっとの変更で輪唱できるようにしておいたぞ。行100の「RIN=0」を「RIN=1」に変更してしてみよう。これでMSXが輪唱する。PSGでは3音まで音を重ねられるからいろいろ試してみよう。

実はPSGでは音程のある音だけでなく、ノイズを発生させることもできる。リスト2を実行すると、海の波の音が聞こえる。またリスト3では、簡単なドラムだけの曲が演奏されるはずだ。PSGはこのように、SOUND文と一緒に使ってノイズを使って効果音を作ることもできたので、いろいろと重宝がられていたんだ。

FM音源を使った演奏

さて今度はFM音源を使ってみよう。とりあえずは、リスト4を打ち込んでみよう。リスト1からちょこっと変更するだけだ。ポイントは行20の「CALL MUSIC」命令と、行1000以降の「PLAY#2,」命令だ。このおまじないでかなり音の雰囲気が変わるぞ。

MMLも拡張されているぞ。オクターブを変更するには今までは「O」コマンドでいちいち指定していたけど、「<」「>」のコマンドで相対的に変更できるようになった。たとえば、"O4L4cdefgabO5ccO4agfedbc"は"O4L4cdefgab>cc<bagfedc"と簡単にできる。また「&」でタイを指定することができる。"c2"と"c8"をひとつの長さにするのは、PSGだけのときはとっても難しいことだったけど"c2&c8"として簡単に1つの音にできるんだ。さらに「{ }n」でカッコのなかの音をn拍で等分することができる。これで3連符など今までやりづらかったことが簡単にできるようになっているんだ。

PSG音源とFM音源とのいちばんの大きな違いが、楽器（音色ともいう）が変えられること。これによりさまざまな音を作ることができる。

音色の指定には「@」コマンドを使用する。たとえばリスト4の行110の"@6"を"@4"とすると、そのパートの音がトランペットからオーボエに変わったと思う。各チャンネルをこんなふうに別々の音に割り当てることもできるんだ。

なお、FM音源を使うと最大でFM音源が9音、PSGが3音の12音同時に出すことができる。

とにかく、習うより慣れろでMML一覧にのっているいろんなMMLを試して、どこを変更したらどんなふうになるかを肌で感じてみよう。

MuSICAを使った音楽演奏

ここまではBASICだけでの話だったけど、ここからは自作ゲームのBGMにするなど高度な音楽の演奏に挑戦してみよう。確かにMMLさえ書ければ、BASICだけでも音楽を演奏させることはできる。でもゲームのBGMにしようとするとなかなか難しい。

ゲームのBGM演奏をさせるには、演奏させるプログラムを処理の速いアセンブラと呼ばれるもので作成することが必要なのだが、これはすごくコンピュータのことがわかっていないと作れない大変なものだった。

そこで、誰もが簡単にゲームに組み込めるように、演奏させるプログラム「ドライバ」というものが開発されたんだ。MSXにはこのような音楽演奏用のドライバはいくつかあるけど、今回はMSXマガジンで一般的に使われていた「MuSICA」と呼ばれるソフトを収録してある。ここからのページでは、このMuSICAを使った音楽の演奏方法を解説していこう。

さて、MuSICAでは音楽作成に関してBASICではやりづらかったり、できなかったりしたことが簡単にできるようになっている。たとえば、音色作り。BASICでも音色を作ることはできるが、値の設定の仕方が面倒なんだ。MuSICAなら直接値を設定するだけで音作りができる。

MuSICAの主な特徴をまとめると、

- 自由に音色が切り替えられる
- MMLが使いやすいように拡張されている
- BASICとは違う独特の編集機能がついている
- デチューンやポルタメントなどの機能を搭載している
- 自作のゲームにBGMとして組み込むことができる

などである。

演奏データの作り方

さっそくMuSICAでの演奏データの作り方を説明していくぞ。まずはMuSICAの起動だ。これはMSX PLAYerのBASIC版を仮想ディス

クAから起動して「RUN "MUSICA.BAS"」と打ち込む。ちなみに終了するにはリセットしかない。操作方法は表1を見てくれ。

MuSICAは3つのモードを切り替えて演奏データを作成する。1つは「テキストエディターモード」で、ここで曲のデータ（ブロックデータやシーケンスデータ）を作成していく。2つ目が「音色エディターモード」で、FM音源やPSG音源の音色を作成することができる。3つ目は「ディスクモード」で、作成した曲データや音色データのロードやセーブを行う。まずはテキストエディターモードから解説しよう。

テキストエディターモード

このテキストエディターモードでは、BASICのようにMMLデータを書いていき、どの順番に演奏させるかを記述していくモード。具体的にはリスト5を見てもらいたい。

まずはブロックデータ部にBASICのときと似た感覚でMMLを書いていこう。MMLはMuSICA用に拡張されている。これは、表2を見てもらいたい。BASICのときとほぼ同じだけど、実はタイ用の「&」コマンドがなくなっている。これは「(」、「)」の「レガートオン／オフ」コマンドで代用する。たとえば"c2&c8"というデータは、"(c2)c8"と書く。

このMMLはBASICでいう文字型の変数に代入するような形で、ラベルを指定してそれにMMLを設定していくんだ。このとき「"」（ダブルクォーテーション）はいらない。BASICに慣れてしまうとよくやってしまうミスなので注意しておこう。ちなみにラベル名はBASICのときのような2文字の制限はないぞ。またコメントはBASICでは「'」（シングルクオーテーション）だったが、MuSICAでは「;」（セミコロン）となる。

このように記述したラベルをどの順番で演奏されるかを記述するのが上部の「シーケンスデータ」部だ。ここではFM1～FM6、FMR、FM7～FM9までのFM音源用のチャンネルとPSG1～PSG3のPSG音源用のチャンネル、そしてSCC音源用のチャンネルがある。これらのチャンネルにラベルを並べることで、どのように演奏させるかを決定する。なお、FMRはリズム音専用のチャンネルだからここに設定できるのはリズム音専用のMMLであるから注意しよう。また、ここにMMLを設定した場合はFM7～FM9にMMLを設定しても音はならないので注意してくれ。それからMuSICAはシーケンスデータを最後まで演奏すると、いちばん始めまで戻って演奏を再開する。曲を終了させるには、リスト5のような最後にしておくといいぞ。

BASICでのPLAY文は横方向にチャンネル、縦方向に演奏する順番を書いているが、MuSICAではそれが逆転していて、縦方向にチャンネル、横方向に演奏順序が書かれている。MuSICAの方が実際の楽譜に近い感じになっているぞ。

さて、通常モードでF5キーを押すと演奏が開始される。エラーが表示された場合は、その場所までカーソルが自動的に移動しているからさくっと修正しよう。聞きたい個所までは［F4］キーで早送りができるのでBASICよりは作業の効率も上がるだろう。

ちょっとしたテクニックだけど、あるトラックをミュートしたい時は、「=」のあとに「;」を書くといい。こうしておけば、あとで元に戻すときも簡単だ。リスト5では、FM音源2チャンネルとPSG音源2チャンネルにこのコメントを入れてある。この「;」を消して演奏してみると、BASICで試したときと同じように輪唱が始まるはずだ。

音色エディターモード

さて次は音色エディターモード。ここでは、FM音源やPSGなどの音色の編集ができる。それぞれのパラメータに関してはページの関係もあって残念ながら書くことができない。FM音源はすべてのパラメータが相互に作用して音色が変化するので、ぜひとも自分の耳でその音を確かめてもらいたい。

「PSG」はエンベロープ、トーンとノイズの切り替え、そしてノイズ使用時はノイズの平均周波数が設定できる。エンベロープの意味は図2を参考にしてもらいたい。SコマンドとMコマンドで設定したものが自由に設定できるようになっている。このエンベロープはVコマンドとは関係なく設定できるので、BASICより使いやすくなっているぞ。BASICでは設定しにくかった演奏中のノイズ周波数の変更などが、MMLの@コマンドで簡単にできてしまうのだ。

表1 MuSICAの操作方法

●テキストエディターモード

キー	機能
F1	テキストの先頭行に移動
F2	テキストの最終行に移動
F3	演奏ストップ
F4	演奏早送り
F5	演奏開始
INS	挿入／上書き の切り替え
DEL	カーソル位置の削除
BS	カーソル前の削除
CTRL＋W	上へ20行スクロール
CTRL＋Z	下へ20行スクロール
CTRL＋E	カーソル以降削除
CTRL＋Y	カーソル行削除
CTRL＋O	カーソル行へ一行挿入
CTRL＋P	カーソル位置へペースト
ESC	コマンドモードへ
L	指定行へジャンプ
A	範囲指定 （指定後）C コピー （指定後）M ムーブ （指定後）T カット
N	テキスト初期化
D	ディスクモードへ
V	音色エディターモードへ

●音色エディターモード

キー	機能
ESC	コマンドモードへ
C	音色のコピー(コピー元番号→コピー先番号)
S	音色の入れ替え(入れ替え元番号→入れ替え先番号)
M	モード変更(FM→PSG→SCCの順に変わる)
Q	テキストエディターモードに戻る
SPCAE	値を＋する
N	値を－する

●ディスクモード

キー	機能
ESC	テキストエディターモードに戻る
1) load MUSIC	テキストデータのロード
2) load VOICE	音色データのロード
3) save MUSIC	テキストデータのセーブ
4) save VOICE	音色データのセーブ
5) save BGM	BGMファイルのセーブ

表2 MuSICAのMML

★MuSICA専用のMML　＊MuSICAで拡張されたMML

	文字	意味	値のとる範囲	備考
★	;（セミコロン）	この後にコメントを書ける		シーケンスデータ用
★	/n	n回繰り返して演奏		シーケンスデータ用
★	",(カンマ)"	ラベルの区切りに使用		シーケンスデータ用
	An～Gn	n分音符の音程の発生	－	
＊	Rn	休符の設定	1≦n≦64	
	Ln	長さの設定	1≦n≦64	
	Tn	テンポの設定	32≦n≦255	
	On	オクターブの設定	1≦n≦8	
	Qn	音の長さの割合	1≦n≦8	
	Vn	音量の設定	0≦n≦15	
	>	オクターブを1つ上げる	－	
	<	オクターブを1つ下げる	－	
	"+,#"	音を半音上げる	－	
	-	音を半音下げる	－	
	.（ピリオド）	音符や休符の長さを1.5倍にする	－	
	"Yr,d"	レジスタrにデータdを書き込む		
★	(	レガートオン		
★	)	レガートオフ		
＊	@n	n番の音色に切り替える	FM 0≦n≦99 PSG 0≦n≦29 SCC 0≦n≦49	
★	Sn	サスティンの指定	OFF 0 ON 1	
★	Zn	デチューンの指定	0≦n≦255	
★	In	ビブラートの深さ	0≦n≦255	
★	Pn	ポルタメントの変化幅	0≦n≦255	
★	Mn	"I,Pコマンドのスピード"	0≦n≦255	
★	Wn	nで設定された長さだけ状態を維持する	1≦n≦64	
	Bn	n分音符バスドラム音を発生	1≦n≦64	リズムパート専用
	Sn	n分音符スネアドラム音を発生	1≦n≦64	リズムパート専用
	Mn	n分音符タムタム音を発生	1≦n≦64	リズムパート専用
	Cn	n分音符シンバル音を発生	1≦n≦64	リズムパート専用
	Hn	n分音符ハイハット音を発生	1≦n≦64	リズムパート専用
★	VBn	バスドラム音の音量	0≦n≦15	リズムパート専用
★	VSn	スネアドラム音の音量	0≦n≦15	リズムパート専用
★	VMn	タムタム音の音量	0≦n≦15	リズムパート専用
★	VCn	シンバル音の音量	0≦n≦15	リズムパート専用
★	VHn	ハイハット音の音量	0≦n≦15	リズムパート専用
	Vn	リズムパートの音量の設定	0≦n≦15	リズムパート専用

「SCC」は波形の形状とエンベロープを設定できる。波形データは初期設定で入っているデータを参考に作ってもらいたい。波形の最初のデータと最後のデータは同じにしておかないとノイズが発生する原因となるので、注意しよう。エンベロープに関してはPSGとまったく同じだ。

なお、テキストエディターモードに戻るにはコマンドモードの「Q」だ。

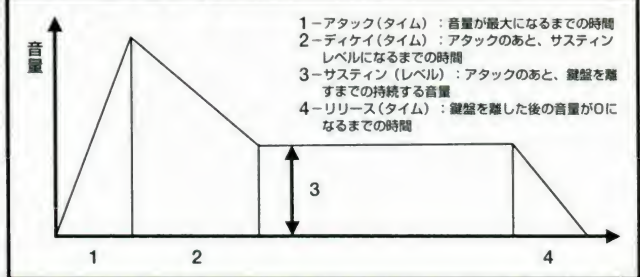
ディスクモード

お次はとっても大切なディスクモードだ。このモードでは、テキストエディターモードや音色エディターモードで作成したデータを読み出したり、保存したりできる。作ったデータは必ずこのモードで保存するように心がけよう。特に音色データは保存するのをよく忘れがちだから注意しよう。

自作ゲームへの組み込み

さて、最後になるけど自分のゲームに組み込む方法を説明しよう。

まずは演奏データを組み込み用のデータ形式に変換しなくてはならない。これはディスクモードの5番で行うことができる。「5) save BGM」を選ぶと保存するデータの先頭アドレスを聞いてくる。よくわからなければそのままリターンキーを押してしまおう。よくわかっている人は自分のゲームの使用しているプログラム領域と重ならないように設定しよう。その後、データのファイル名を聞いてくるので、適当につけてしまおう。

ここまではMuSICAでの作業だがこのあとは、自作ゲームでのプログラムの追加だ。これはリスト6を参考にしてもらいたい。絶対に必要なのは、ドライバの初期化で、ここではサンプルリストの行100、110で行っている。あとは行130以降のように、演奏回数とBGMデータのアドレスを指定してBGMをスタートさせればいい。演奏回数を0にすると、無限ループする。ディスクアクセスするときは、BGMをストップさせておこう。これは行190以降の処理だ。

行220以降はボリュームの調整だ。ここを応用してフェードイン・アウトも作れる。なお17を指定すると一時停止、18で再開となっている。

ここまですごく簡単にBASCとMuSICAでの曲つくりのことを書いてきたけど、本当はほかの人のプログラムを見て真似するのがいちばんの勉強になる。BASICについては、4曲本格的なサンプルを収録しているので、いろいろといじってみてほしい。

図2 ADSRの意味

リスト5

```
;LIST 5 ｶｴﾙ ﾉ ｳﾀ (MuSICA)
FM1=MAIN,A0,A1,A2,A3,A4,END/255
FM2=;SUB,A0,R,A1,A2,A3,A4
FM3=
FM4=
FM5=
FM6=
FMR=
FM7=
FM8=
FM9=
PSG1=MAIN,A0,A1,A2,A3,A4
PSG2=;SUB,A0,R,A1,A2,A3,A4
PSG3=
SCC1=
SCC2=
SCC3=
SCC4=
SCC5=
```

音源のトラックにレベルを設定する（FM1行）
FM1～FM9：FM音源シーケンスデータ（演奏順序の定義）
PSG1～PSG2：PSG音源シーケンスデータ
PSG3～SCC5：SCC音源シーケンスデータ

```
R=z32 rrrr

MAIN=@6 O5
SUB=@4 O6

A0=T120 L4 V14
A1=cdef edcr
A2=efga gfer
A3=crcr crcr
A4=L8 ccddeeff e4d4c4r
END=r1
```

MAIN～END：ブロックデータ（ラベルの定義）
ラベルにMMLを設定する（A4行）

リスト6

```
1 'LIST 6 BGM ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
10 CLEAR100,&HC800:DEFINTI:DIMI(1)
15 BLOAD"BGM.BIN"
20 BLOAD"TEST.BGM"
30 GOSUB100:V=0
40 A$=INPUT$(1)
50 IF A$="0" THEN GOSUB130
60 IF A$="1" THEN GOSUB190
70 IF A$="-" AND V<>15 THEN
V=V+1:GOSUB220
80 IF A$="+" AND V<>0  THEN
V=V1:GOSUB220
90 GOTO 40
99 '--------------------------------
100 DEFUSR=&HCE00          'BGM ｲﾆｼｬﾗｲｽﾞ
110 I=USR(0):RETURN
120 '--------
130 I(0)=0                 'ｴﾝｿｳ ｶｲｽｳ
140 I(1)=&HC800            'BGM DATA ｱﾄﾞﾚｽ
150 DEFUSR=&HCE03          'BGM
160 I=USR(VARPTR(I(0)))
170 RETURN
180 '--------
190 DEFUSR=&HCE06          'BGM ｽﾄｯﾌﾟ
200 I=USR(0):RETURN
210 '--------
220 I(0)=V                 'VOL.(0-15)
230 DEFUSR=&HCE09          'ﾏｽﾀｰﾎﾞﾘｭｰﾑ
240 I=USR(VARPTR(I(0)))
250 RETURN
```

● 収録曲はコレだ！ ●

今回のBASIC用音楽プログラムをアレンジ1曲にオリジナル3曲の4曲用意したぞ。再生するにはディスクAからそれぞれのプログラムをLOAD後、RUNしてくれ。「リンゴの唄」と「DEEPMARINE FOREST」は曲が終わるとコマンドモードに戻るけど、後の2曲は永久ループするので、適当なところで[CTRL]＋[STOP]キーで終了だ。プログラムを作成した作曲者の皆さんにコメントをもらってあるので紹介しよう。曲の解説は、よっちゃんこと横川理彦さんだ。

作者：JL2TBBさん
曲名：リンゴの唄
ファイル名：APPLE.FM
作者コメント：私の手元にはもうMSX-BASICを実行できる機械がありません。壊れたあと、引っ越しのため廃棄してしまったのです。ですから、公式エミュレータ登場というのは非常に喜ばしいですね。今回プログラムを提出させていただくことになりましたが、MSX-MUSICの資料さえも残っていないので、昔作成したプログラムをそのまま投稿しました。
解説コメント：「リンゴはとってもロンリー・ハート」と名付けたいダブルミュージック（違う曲を同時に演奏すること）。音の感じは鯖ミソにアップルジャムのトッピングした感じでは？　今、巷では80年代のデジタルが来ている……といわれても、ちょっと腰が引けるなあ。

作者：yambalさん
曲名：DEEPMARINE FOREST
ファイル名：DEEP.FM
作者コメント：yambalと書いてGANPと読む(？)28歳です。近頃は素数変拍子で自分に挑戦することに凝っていまして、この曲は11/16拍子。約10年ぶりのMSXでしたが、この環境は可能な全てのパラメタを心ゆくまでいじり倒せるところに魅力がありますね。自サイトはhttp://www.f-kid.com/です。
解説コメント：4分近い力作、変拍子バリバリです。立ち上がりの鋭いヴィブラフォンと、遅いブラスの対比がうまく海＆森のアンビエント感につながっている。この頃のお安い「癒し系」音楽を思い切り貶したい私ですが、元GANP28歳のこの曲でちょっとほっとしたのだった。

作者：MABさん
曲名：Hammer Game
ファイル名：HAMMERG.FM
作者コメント：ハニリイト。3年前にMIDIで作った曲を移植したものです（MSX時代の作風に近い物をチョイスしました）。モグラ叩きをイメージし目まぐるしさや高揚感を強調しております。モグラを見たことないですが。
解説コメント：やた！　フュージョン満開のウルトラ・テクニックものですな。リズム、コード、曲構成、いずれも抜かりのない優秀作だ。サビでシンセ×2の追っかけあいになるところに高揚感がある。FM音源にはフュージョンとへなちょこ＆カックンものが良く合う。

作者：K.N.
曲名：ハイロウ
ファイル名：HILOW.FM
作者コメント：なんとも言えないヘチョさを表現してみたくて作りました。特にイントロで「うわ！　だっさー！」と思っていただければ、まさに作者の思惑通りです。で、メインのループに入って急に曲調が変わりますが、これは単純に安っぽさを目指してみました。「カッコイイ(と多くの人が思うような)音楽」以外にも音楽の良さって色々ありますよね～、という意味を込めています。楽曲というよりはサウンドロゴ的な感じですね。
解説コメント：とかいってたら(Hammer Game参照)出た！　カックンものです。つっても、技術的にも上手い。間抜けなフレーズを組み合わせて飽きないように構成していくのはセンスが必要。個人的にはループ始めあたりの「木曽のなー」と聞こえる民謡フレーズがお気に入りです。

実機MSXを再検証

MSX解体新書

MSXは、1983年に仕様が発表され、それから10年以上ユーザーに親しまれたパソコンだ。それまでパソコンというと、今のように誰の家にもあるというものではなく、まだまだ一部の人のものだった。そこに「ホームコンピュータ」という理念をひっさげて華々しくデビューしたのがMSXだ。今はホームコンピュータというと、WindowsやMacがそれにあたるが、その元祖といってもいいだろう。しかし、MSXが発売されなくなって、すでに10年近い年月が経過し、今となっては「MSXってなに？」という言葉さえ聞こえるぐらいだ。そこで、MSXがどんなパソコンなのか知らない人のために軽くおさらいしておこう。

TEXT：高橋秀樹、ほか

西暦	その年の出来事
1983年	●MSX(1)の仕様発表
1984年	●各社MSXマシン出揃う（松下、ソニー、サンヨー、東芝、日立、三菱、ビクター、ヤマハ、パイオニア、キヤノン、ゼネラル、富士通）
	●ソニーと東芝から1DD（片面倍密度精度）タイプの外付けフロッピーディスクドライブが発売（8万9800円）
	●カシオから2万9800円の低価格MSXが発売
	●ソニーからMSX初のディスク内蔵マシンが登場
1985年	●MSX2の仕様発表
	●東芝から初のMSX2マシンが発売
	●MSX100万台キャンペーン（新宿に巨大な恐竜出現）
1986年	●NEOSからMSX1を2にするバージョンアップアダプタ登場
	●パナソニック（FS-A1が29800円）とソニー（HB-F1が32800円）から低価格MSX2が登場
	●パソコン通信の「アスキーネットMSX」が運営開始（現在は終了）
	●アスキーがメガロムの仕様を発表
	●アスキーから日本語変換システム「MSX-JE」内蔵のワープロ「MSX-Write」が発売
1987年	●通信モデムを内蔵したMSX2登場（ソニーのHB-T7は1200bpsのモデム搭載）
	●ディスク内蔵型の低価格マシン登場（ソニーのHB-F1XDと、パナソニックのFS-A1Fが、それぞれ5万4800円で登場）
1988年	●アスキーがMSX-DOS2発売
	●MSX2+の仕様発表
	●パナソニック、ソニー、サンヨーからMSX2+マシンが発売
1989年	●アスキーからMSX HD Interfaceが発売
	●ソニーからビデオデジタイザーが登場
1990年	●MSX出荷台数が国内外で400万台を突破
	●MSX turboR仕様発表
	●パナソニックからMSX turboR仕様のFS-A1STが発売
1991年	●パナソニックからMSX turboR仕様のFS-A1GTが発売
1994年	●最後のMSX実機となるFS-A1GTが生産中止
	●最終的にMSXは500万台近い出荷数となる

※年表はMSXマガジン100号に掲載された記事を元にしています。

8ビットパソコンのPC/AT互換機

振り返ってみると、MSXとは実にいろいろな顔をもったパソコンだった。あるときにはゲームマシンであり、あるときにはソフト開発ができ、あるときは音楽演奏やビデオ編集に特化した機能が備わり、あるときは優れたワープロが登場した。むろんパソコンがいろいろな用途に使えるのは当たり前だが、MSXの場合、他の8ビットパソコンに比べても遥かに柔軟に対応できる懐の広さがあった。

そもそもMSXの規格を提唱したのはアスキー／マイクロソフトだったが、実際に製造販売したのは、松下、ソニー、サンヨー、東芝、日立といったメーカーだ。つまり、MSXという規格を元に、メーカーは独自の機能やカラーを打ち出したマシンを発売していたわけだ。多くのメーカーで同じ規格のマシンを作る意味は果たしてあったのか？　当時はそんな声が常にあったようにも思う。しかし今、多数のメーカーが参入しているPC/AT互換機の状況を見れば、間違いではなかったといえる。各メーカーから発売されたマシンは統一規格の元にさまざまな個性を放ち、ユーザーも自分の夢を実現してくれる最適なマシンを選ぶことができた。MSXはいわば8ビットパソコンのPC/AT互換機だったのだ。

家庭を意識した真のホームパソコン

MSXは「家庭」を強く意識したマシンだった。第一に、MSXは普通のテレビに接続して使うことができた。他の8ビットパソコンは軒並み解像度を高くしていったため、グラフィックスは美しかった半面、専用ディスプレイを必要とした。第二に、MSXはカートリッジをガチャンと差してソフトを動かすことができた。他の8ビットパソコンはゲームをするときでさえBASICの知識が要求された。そして第三に値段が安かった。他の8ビットパソコンが軒並み10～30万円ほどしていたのに対し、MSXは5～10万円程度がメインの価格帯だった。他にも、家電メーカーが作っていることの安心感や、MSXのマークがついたソフトであれば動作するという分かりやすさの点もあった。

むろん、これらの特徴の一部は「MSXはキーボードのついたゲームマシンだ」と評されるネタにもなったが、いずれにしてもMSXは、従来とは違う「新しいユーザー」を確実に開拓していった。すなわち、コンピュータのマニアではなく、一般の人がコンピュータに触れられる環境を提供したのだ。こうして、MSXという画期的なマシンは「ホームコンピュータ」の道を歩んでいくのである。

MSX本体
MSX BODY

MSXは決して１つの場所にとどまっていたわけではない。時代の要請に合わせて3度のバージョンアップを遂げている。それぞれ、どのような特徴があったのか紹介しよう。

MSX1

MSX1は、パソコンにはじめて「統一規格」という発想を持ち込んで1983年に複数のメーカーによって登場した。発売されたMSXは70機種以上。こうした動きは世界的にみても他に類はない。当初は、メーカーごとにどれだけの特徴が出せるものだろうかと心配されたが、実際には、マシンのデザイン形状はもちろんだが、RAM容量、スロット数、キー配列の違いによりバラエティ豊かな製品がラインナップされた。また、よりコンセプトを明確にした機種も現れた。ゲーム機に特化させて値段を下げてきたマシン、ミュージック機能を搭載させたマシン、ビデオコントロールをできるようにしたマシン、などだ。価格の安さと扱いやすさから一般に広く支持され、1985年には出荷100万台を達成した。

MSX2

1985年5月に、MSX1の上位互換をうたって発表されたのがMSX2だ。この規格では、MSX1で不満だった多方面の拡張がなされた。まずRAMが、標準で64Kバイトに増加。プログラミングもたっぷり楽しめるマシンとなった。

そして忘れてはならないのが新VDPのV9938。この搭載により、それまで貧弱といわれていたグラフィックス機能が一気に向上。色数や解像度が大幅に改善された。CLOCK-ICも初めて組み込まれ、バッテリバックアップによるカレンダーと時間の保存も可能になった。そして、これらをBASICからもサポートできるように、バージョンが1.0から2.0へとアップした。

形状的には、当初発売された機種こそキーボード一体型が多かったものの、やがて高級機が現れ、セパレート型が普及する。そして87年の秋には、逆に松下電器とソニーから3万円を切る一体型のマシンが発売。一気に普及率を押し上げることにもなった。

MSX2+

MSX2から、さらにグラフィック機能を中心に強化が図られたのが1988年に登場したMSX2+だ。VDPがV9958になり、MSX2では最大同時256色表示だったが、MSX2+では一気に19268色まで向上した。これは、YJKモード（自然画モード）と呼ばれ、1ドットあたり8ビットで情報を持ち、連続した4ドットを一組として処理するというものだった。それと、ゲーム用には、MSX2で加わった縦方向のハードウェアスクロールに、横方向も追加された。

MSX2+では、漢字ROMが標準搭載されており、BASICからも漢字を使うことができた。

このMSX2+によってMSXの機能的な不満はほぼ解消され、ある域での完成をみたといっていいだろう。とはいえ、発売されたMSX2+機は7機種程度だった。

左からソニーのMSX2+機「HB-F1」、松下電器のMSX turboR機「FS-A1GT」（右上）と、三菱電機のMSX2機「ML-TS2H」（右下）。それぞれが個性を放っている。

MSX turboR

1990年10月15日に発売された、MSX turboRには「R800」が搭載された。その恩恵は、まず速度に現れ、Z80CPU換算で29MHz程度と、従来の数倍のスピードをたたき出した。

CPUの高速化によって、大容量のデータが扱えるマシンとなったため、メインRAMも256Kバイトまで拡張されている。このパフォーマンスを活かす機能も同時に追加されている。

音楽面ではPCM録音再生機能を追加。8ビットサンプリングでハイクオリティな音声が取り込めるようになった。また、MSX-ViewというGUIも発売され、Windowsのようにマウスを使った環境が利用できるようになった。

まさにMSXは、現在のようなホームパソコンになろうとしていたが、発売されたMSX turboR機は松下電器の2機種のみ。1994年には最後のMSX turboR機が生産中止となり、MSX実機の時代は終わった。

MSXスペック表

	MSX	MSX2	MSX2+	MSXturboR
CPU	Z80A（3.58MHz）	同左	同左	Z80+R800（7.6MHz）
ROM	システムROM（32KB）	システムROM（48KB）	システムROM（80KB）	システムROM（160KB）
RAM	8KB以上	64KB以上	同左	256KB以上
VRAM	16KB	64KB/128KB	128KB	同左
VDP	TMS9918	アスキー／ヤマハV9938	アスキー／ヤマハV9958	同左
最大発色数	16色	256色	19268色	同左
最大画面解像度	256×192ドット	512×424ドット（インターレース時）	同左	同左
スプライト	モード1	モード1/モード2	同左	同左
スクロール	—	縦	縦横	同左
PSG音源	8オクターブ3重和音（AY-3-8910相当）	同左	同左	同左
FM音源	オプション	オプション（同左）	同左	MSX-AUDIO（YM8950） MSX-MUSIC（YM2413）
PCM音源	—	—	—	8ビット（モノラル）
MIDI IF	—	—	—	オプション
FDD	オプション	同左	同左	2DDドライブ1台以上
HDD	オプション	同左	同左	同左
メモリーディスク	—	あり	同左	同左
映像出力	RF、ビデオ、アナログRGB	同左	同左	同左
プリンタ	オプション	セントロニクス社準拠	同左	同左
スロット	1スロット以上	2スロット以上	同左	同左
ジョイスティック	1または2ポート	2ポート	同左	同左
カセットIF	1200/2400ボー	同左	同左	—
内蔵時計	—	あり	同左*	同左
BASIC	MSX-BASIC ver1.0	MSX-BASIC ver2.0	MSX-BASIC ver3.0	MSX-BASIC ver4.0／4.1
MSX-DOS	MSX-DOS(オプション)	MSX-DOS1/2(オプション)	同左	MSX-DOS1/2

CPU
CENTRAL PROCESSING UNITS

計算をしたり、さまざまな制御を行ったりと、パソコンの重要な部品がCPUだ。すべてのMSXにはZ80CPUが使用され、そしてMSX turboRにはR800が搭載されている。

Z80

MSXで使われているCPUは、ザイログ社のZ80だ。これは、NECのPC-8001/PC-8801/PC-6001、シャープのMZ-80、X1シリーズなど、MSX以前にも多くの機種で採用されており、8ビットパソコンのスタンダードと言えるCPUである。現在でもそうだが、メーカーがパソコンを作る場合、比較的新しいCPUを選択したくなるものだ。しかし、MSXでは奇をてらうでもなく、非常にありきたりなCPUが選ばれた。そして、そのことがMSXに多大なる恩恵をもたらした。

まず、Z80に精通している人が多く、情報も膨大にあり、プログラマが比較的早くMSX用のソフトウェアの開発に打ち込むことができた。これは知識だけの問題ではなく、プログラムの開発ツールもすべて最初からそろっていたわけである。また、同じCPUで開発されたソフトウェアに関しては、場合によっては画面に関するルーチンだけを書き換えるだけで移植が可能だった。こうしてMSXのソフトウェアは一気に増えていくことになるのだ。

また、Z80自体、ある意味で楽しみの多いCPUでもあった。裏レジスタという補助的な一時メモリがあったり、未定義命令という一般公開されていない命令が存在するなど、こうした技巧的な部分がマニア心をくすぐった。実際、これらを利用することで、速度を上げたり、プログラムを短くすることができたのだ。そんなことからも、Z80に愛着を持っていた人は多かった。

R800

MSXからMSX2+に至るまで、CPUは手付かずのままだった。CPUを変更することは、人でいえば頭脳を取り替えるような行為。それは、姿形は似ていても別人格となってしまう。MSXにおいても、CPUの変更は、これまで築き上げてきたものをすべて壊しかねない領域である。よほどの決心がなければ、おいそれと手を出せるものではない。

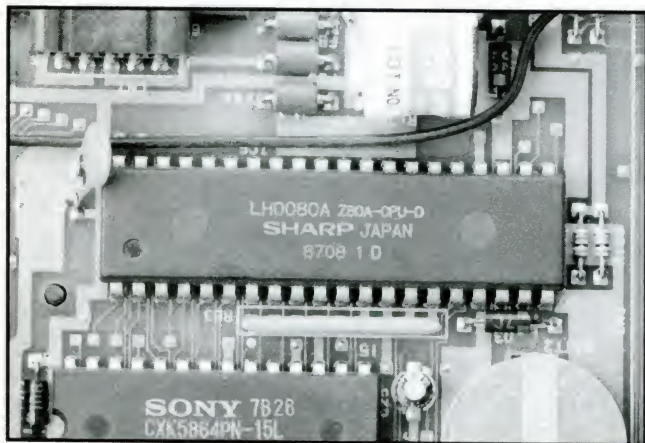
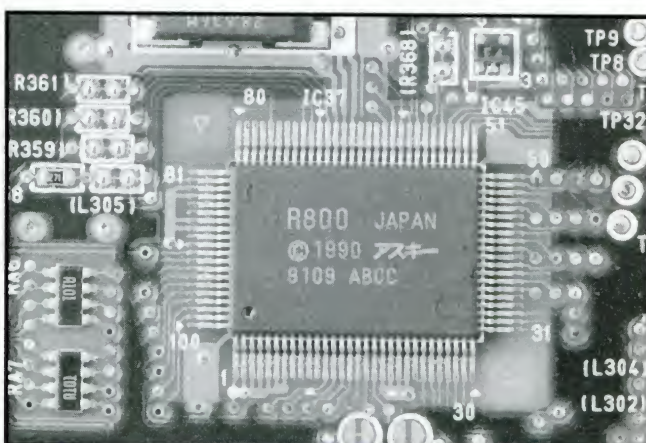
しかし、より大きなデータを扱うためには、CPUへのてこ入れが必須だ。そこでアスキーがMSXのために開発したのがR800という16ビットのRISC CPUである。ちなみにRISCとは命令セットを縮小して高速化をねらったCPUのこと。Z80のCISCと呼ばれるコンピュータ・アーキテクチャに比べると、格段の速度向上が望めるはずである。

アスキーは、R800の開発を考える前に、むろん他のCPUについても検討を重ねていた。しかし、決定的な代替CPUは見つからない。そんななか、アスキーのエンジニアの1人が趣味でZ80アーキテクチャのCPUを設計しており(むろんMSXの今後の開発に役立てばと思ってのことだが)、それが使えそうだということで新CPUの開発に踏み切った。

R800は、Z80のソフトウェア上位コンパチブルな高速CPU。Z80用のソフトウェアを、そのままR800で高速実行できる。Z80に追加された命令は16ビットの乗算命令と、Z80で裏技的に使われていたIX/IYレジスタのバイトアクセス命令など。内部処理は16ビットで行っているが、外部とのデータのやりとりは8ビットという変則的なもの。アドレスも8ビットで管理されていたため、アクセスできるのは64Kバイト。したがって、MSX turboRはRAMを標準で256Kバイト搭載されてはいたが、従来のメモリーマッパーという概念でメモリアクセスが行われることになった。

R800はZ80の約29MHzに相当する速度であり、実際のテストでもMSX2+に比べて平均でも4～5倍、最大で10倍のスピードアップが実現されたのだ。

ところでMSX turboRには、なんとZ80も搭載されている。R800にはZ80のコードも実行できるのだが、これはなぜか。実は、R800でMSX1～2+用のゲームを動かしてみたところ、速すぎて遊べないことが判明。遅くするための開発には経費がかさむ。そこで、コストが安くなっていたZ80も搭載し、デュアルCPUになったというわけだ。

ちなみに、MSX turboRではZ80とR800をプログラムで随時切り替えられる。こうしたしくみで、MSX turboR専用ソフトを使ったときには、すぐ高速なMSXに変身するのである。

すべてのMSXに搭載されているZ80がこれ。動作クロックは3.58MHz。MSX以外にも多くの機種で使われてきたCPUである。

アスキーが作った16ビットRISC CPUのR800。これのおかげでMSX turboRは、以前のMSXよりも4～10倍高速なマシンに変貌した。やや時代に取り残された感のあったMSXユーザーにとって喜びもひとしおだった。

VDP
VIDEO DISPLAY PROCESSOR

MSXの大きな特徴の1つがVDPを搭載していたことだ。このチップによって、MSXは他のパソコンとは一線を画すグラフィックスの世界を切り開くことになる。

VDP

MSXではVDP（Video Display Processer）と呼ばれる専用のチップが画像まわりを一手に引き受けていた。そのためメインのCPUはVDPに命令を送るだけで、あとはVDPが自動的に画像処理を行ってくれた。しかもVDPとVRAMは直接つながっていたため、その中で閉じており、メインメモリに影響を与えることもなかった。

ちなみに、MSX1に採用されていたVDPはテキサス・インスツルメンツ社のTMS9918という汎用品。セガのゲーム機「SG-1000/1000 II」「SC-3000」などでも利用されていた。MSX2にバージョンアップするにあたり、アスキーとヤマハによってV9938を開発。MSX1の時代にはVRAMが16Kバイトしかなかったが、MSX2/2+では128Kバイトに向上している。そして、このVRAMを使ってMSX2では256色、MSX2+では19268色の同時発色が実現されている。

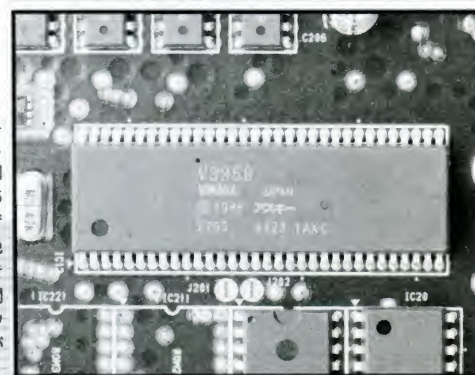
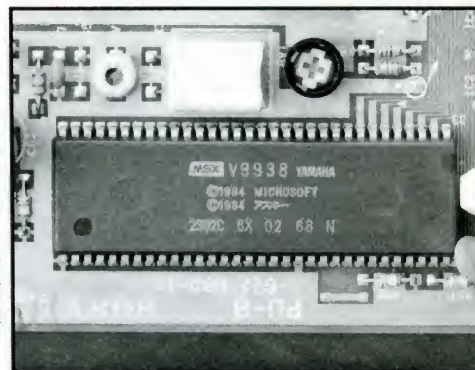
VDPによって、メインのCPUは一切の画像処理から開放され、MSXの構造自体が非常にすっきりしたものとなった。また、画面に関する機能を向上させる際にも、CPUやメインメモリに気を使うこともなく、単にVDPより先の部分だけを拡張すればよいことになった。考えてみれば、これはPC/AT互換機のビデオカードと同じ概念と言える。

VDPにはさらに特徴があった。それは複数の画面モードの存在だ。MSXには4種類の画面モードが、MSX2には9種類の画面モードが、そしてMSX2+からは実に12種類の画面モードが備わっている。ゲームに向いたモード、漢字表示に向いたモード、あるいはグラフィックスに向いたモードなど、これによってMSXの適用範囲はぐっと広がった。さらにVDPを駆使すると、ハードウェアによるスムーズスクロールができたりと、楽しい機能が満載。いじりがいもたっぷりだった。

参考までに、当時の8ビットパソコンの多くは、メインメモリの一部をビデオメモリ（VRAM）に割り当てて、メインのCPUによってすべて制御していた。画面モードは640×400ドットといった高解像度グラフィックスモードのみ。優れもののグラフィックス画面は、静止画や漢字表示などには向いていたが、ダイナミックな動きをするゲームなどでは若干つらいところがあったといえる。

MSX2に搭載されている「V9938」。TMS9918の完全上位互換性を持つVDPだ。画面モードが拡張され、64Kバイトまたは128KバイトのVRAMを使って、美しいグラフィックスを披露した。

MSX2+に搭載されている「V9958」。V9938に機能追加を図ったVDPである。自然画モードでは、最大19268色の同時発色ができた。また、ゲーム開発者には、ハードウェア横スクロールができたのも魅力。

スプライト

ゲームになくてはならないのがVDPの一機能であるスプライトだ。これは、アニメのセルのように背景画像と独立した画像をハードウェアで表示するというもの。ゲーム専用機ではお馴染みの機能だ。これを使うと、滑らかに、しかも高速にキャラクタが動かせるとあって、多くのユーザーをプログラムの道に引き込んだのである。

スプライトを使うには、まずスプライトパターンを用意する。後はXY座標を指定するだけでよい。これで画面の指定位置に瞬時に表示される。そしてXY座標を少しずつ更新すれば、スプライトは滑らかにある方向へ移動する。それがインベーダーのデザインであれば、キャラクターが画面内を動き回るのだ。まさにゲームにはもってこいの機能といえる。

スプライトにはモードが2つ存在する。モード1はMSX1からのもので、単色ながら16×16ドットサイズなら1画面に8枚（8×8ドットサイズなら32枚）が表示できる。ただし、横に4枚以上並ぶと5枚目は表示されないという制約が存在するのが泣き所だ。

MSX2以降になると、モード2が追加される。1画面に表示できるスプライト数は同じだが、制約が少し緩和され、横に8枚まで表示できるようになった。また、複数のスプライトをグループにすることで、ドット単位で色をつけることも可能となった。MSX2以後は、こうしてカラフルなキャラクタが動くようになったのだ。

スクリーンモード

モード	文字数	ドット	色	パレット	スプライト	備考
SCREEN 0	40×24	—	512色中2色	0～15	—	1行40文字のテキスト。文字色は1色
SCREEN 0	80×24	—	512色中4色	0～15	—	1行80文字のテキスト。文字はブリンク可能
SCREEN 1	32×24	—	512色中16色	0～15	モード1	1行32文字のテキスト。色つき文字が可能
SCREEN 2	32×24	—	512色中16色	0～15	モード1	テキスト文字8ドット単位で色指定可能
SCREEN 3	—	64×48	512色中16色	0～15	モード1	1文字を4ブロック分割した模擬グラフィック
SCREEN 4	32×24	—	512色中16色	0～15	モード2	スプライトモード2が使えるSCREEN 2
SCREEN 5	—	256×212	512色中16色	0～15	モード2	ビットマップグラフィックスモード
SCREEN 6	—	512×212	512色中4色	0～3	モード2	ビットマップグラフィックスモード
SCREEN 7	—	512×212	512色中16色	0～15	モード2	ビットマップグラフィックスモード
SCREEN 8	—	256×212	256色中256色	—	モード2	ビットマップグラフィックスモード
SCREEN 10	—	256×212	12499色	—	モード2	YJK/RGB/自然画モード
SCREEN 11	—	256×212	12499色	—	モード2	YJK/RGB/自然画モード
SCREEN 12	—	256×212	19268色	—	モード2	YJK/自然画モード

注：SCREEN9はハングル用のモードであり日本では未使用

スロット
SLOT

MSXに必ずあったのが、スロットと呼ばれる差し込み口だ。ゲームソフトばかりではなく、さまざまな周辺装置が接続でき、MSXを簡単に拡張できたのだ。

スロット

MSXにはカートリッジスロットが必ずある。1つの機種もあったが、2つが標準的。中には3つの機種もあった。ここは、ファミコンのようなゲーム機と同様に、「ゲームカートリッジを差し込むところ」という認識が主だった。しかし、MSXの場合にはさまざまな拡張機器を接続することもできた。具体的には、モデムカートリッジ、FM音源カートリッジ、フロッピーディスクドライブといったものだ。スロットを使うと、PC/AT互換機のスロットと同様に、簡単に機能拡張を行うことができたわけだ。

ハードウェアから見るとスロットとは、本来64Kバイトしか扱えないZ80で最大1Mバイトのメモリを扱えるようにする構造を言う。具体的には、まず64Kバイトのメモリ空間が4本分あり、それぞれを「基本スロット0～3」と呼び、64Kバイトのメモリ空間を4分割したそれぞれを「ページ0～3」と呼ぶ。CPUは各ページに対してアクセスしたいスロットを選択し、これで64Kバイト×4のメモリにアクセスする。たとえば、ページ0はスロット1、ページ1はスロット2、ページ2はスロット1、ページ3はスロット3、という感じである。また、基本スロットからは拡張スロット0～3が拡張できるため、基本スロット4×拡張スロット4＝16スロット＝1Mバイトのメモリ空間にアクセスできるわけだ。

スロットはメモリそのもののすげ替えを意味する。したがって、ゲームカートリッジを差せば、メモリにはゲームプログラムが現れ、ゲームが遊べるようになる。FM音源カートリッジが差されれば、メモリにはFM音源の入出力ポートと、FM音源の制御で使用するソフトウェアが現れる。こうしてMSXはさまざまな姿に変身した。

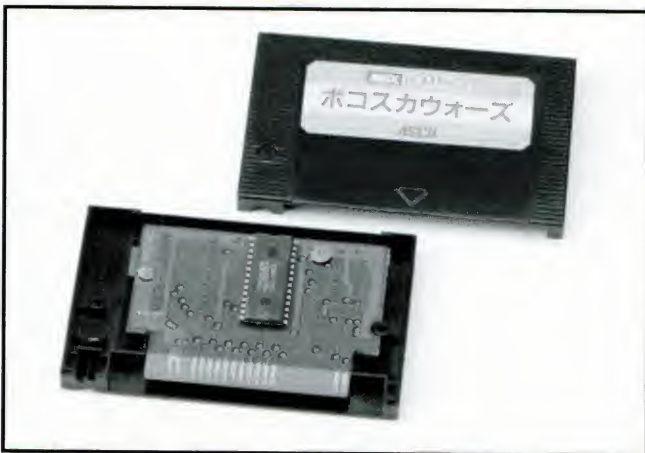
現在、Windowsでもプラグ&プレイという考え方が普通になっているが、MSXの場合も差せば使えるという点で、すでにプラグ&プレイを実現していたと言えるのである。しかも、MSXの場合ドライバのインストールなども不要という点では、ある意味先を行っていたのかもしれないのだ。

カートリッジスロットのコネクタ。ここにROMカートリッジやインターフェイスカートリッジが差さるのである。全部で50ピンあり、アドレスバスとデータバスがすべて並んでいる。

メガロム

当初ゲームカートリッジに使用されたROMは16Kもしくは32Kバイト程度のサイズだった。しかし、よりおもしろいゲームを作るには、さらに多くのメモリが必要だ。そこで登場したのがメガロムという規格のカートリッジである。メガロムには1M～8Mビット（128K～1024Kバイト）の容量があり、それまでの数倍の容量が確保できた。

初めてのメガロムゲームはコナミの「グラディウス」。大容量を活かして、オリジナル版の全ステージが移植され、さらにMSXオリジナルのステージも入っていた。その後は、300種類に近いほどのメガロムによるゲームカートリッジが登場。ところが、やがて半導体不足で満足な供給が得られなくなり、その後安価なディスク内蔵マシンが出現したことによって、次第にディスク版のソフトが増加。メガロムソフトは減少していくことになった。

MSX-DOS2カートリッジと、松下電器から発売されたFM音源カートリッジ「FM-PAC」。MSXのスロットにはシステムを拡張する製品がいろいろあった。なお、右下は参考までのファミコン用カートリッジ。

ゲーム用のROMカートリッジを分解したところ。基盤にゲームのROMが仕込まれている。これを差し込むことによって、メモリにゲームROMが現れ、MSXはゲームマシンと化すのだ。

サウンド機能

SOUND

MSXのサウンド機能は、最初はごく一般的なPSGが搭載されていた。やがて音楽にこだわりを見せるメーカーによってFM音源が登場し、仕様として定着するのだ。

PSG音源

MSXにはサウンド音源が標準搭載されている。それが米General Instruments製の「AY-3-8910A」だ。このチップはPSG（Programable Sound Generator）と呼ばれ、3音＋ノイズを同時発生する機能がある。同チップは、ジョイスティックの入力を受け持つ汎用I/Oとしての機能も有しているため、その利便性から当時のパソコンやゲーム専用機によく採用されていた。音質的には、かなり貧弱と言わざるをえないし、場合によってはかなり耳障りな音だったりもする。しかし、単純さゆえ、逆に極限まで利用してやろうという試行錯誤がなされ、エコーをかけたり、音声を再生するといった技術が開拓されていく。

PSGが便利だったのは、BASICでMMLというマクロ言語でデータ作成ができたこと。現在のDTM環境とは比べるべくもないが、かつてはコンピュータで音楽というと、こうした形態が一般的だった。

そんな1987年のこと、ゲームメーカーのコナミがSCCという音源チップを開発し、「グラディウス2」や「パロディウス」などの自社ゲームに搭載してユーザーに大衝撃を与えた。性能的にはPSGの強力版という感じで、5音が同時発生可能。キレのよい音質とイカしたアレンジによって、ファンが急増していった。

FM音源

MSX2時代の後半には、FM音源が登場。まず1986年にFM音源とADPCM音源が搭載された「MSX-AUDIO」規格に準ずるカートリッジが発売された。しかし、そのゴージャスさがあだになり高価になりすぎた。また、カートリッジ形態が特殊でスロットに入らない機種があったことから普及には至らなかった。その反省点を踏まえてMSX2+時代に入る直前にFM音源のみにダウンサイジングを図った「MSX-MUSIC」規格が登場。ヤマハ製の「OPL（Y-8950）」の廉価版「OPLL（YM-2413）」（2オペレータFM6音＋リズム5音中3音）が使用されており、1988年夏にSRAM＋FM音源という構成で松下電器から発売された「FM-PAC（Pana Amusment Cartridge）」カートリッジが爆発的にヒット。MSX turboRではついにMSX-MUSIC規格が標準仕様として昇格し、さらにPCMの録音再生機能も追加され、MSXのサウンド機能は頂点を極めるのだった。

また、MSX turboRの「FS-A1GT」には、MIDIが搭載され、外部MIDI音源のコントロールがBASICからできるまでになった。

ディスク（記憶装置）

DISK

MSXが登場したとき、世の中は3.5インチFDDの次代に突入していた。MSXにとって、FDDは高価なものだったが、やがてそれも標準的なデバイスとなっていくのである。

FDD

MSX規格が検討されていた当時は、他のパソコンでは5インチFDDから3.5インチFDDへの転換時期。そのため、MSX用としては最初から3.5インチFDDが検討された。各社のMSX1機が出揃った1984年に、ソニーと東芝から初の外付けFDDを発売。このときの定価は8万9800円。MSX本体並の値段とあって、それほどの普及はしなかった。その後、ソニーのHB-701FDや松下のCF-3300といったFDD内蔵マシンも登場したが、高価だったため、使える人だけの特権のようなものだった。しかし、1987年にソニーのHB-F1XDと、パナソニックのFS-A1Fという2機種のMSX2が登場。これらはディスクが付いて5万4800円という安さだったため、急激にFDDのシェアを広げることになる。

MSXのFDDは、当初は片面倍密倍トラック（360KB）の1DDタイプのみ。しかし、すぐに両面倍密倍トラック（720KB）の2DDタイプも発売された。

MSXのディスクのフォーマットはMS-DOS互換となっている。このおかげで、今でも2DD対応のPC/AT互換機で読み出せる。このことは、MSXが今なお陳腐化しないひとつの要素となっている。

MSX2+の時代には、ハードディスクの要求も高まり、MSX-DOS2の発売とともに、ハードディスクインターフェイスなども発売された。しかし、HDDが高価だったこともあり、この時期に恩恵を受けた人はわずかだった。

Quick Disk

ユニークな記録メディアとしてQuick Diskというものがあった。これは、2.8インチのメディアに、片面64KB（両面128KB）のデータが記録できる。アクセスがシーケンシャルであったため、高速なカセットテープという感じで使うことができる。しかし、DiskBasicがユーザーメモリを減らしてしまうのに対して、QuickDiskはインターフェイス内にワークエリアを持っており、ユーザーメモリが減らなかった。そのため、大きなプログラムが実行でき、一部には好評だった。

ヤマハの外付けタイプのフロッピーディスクドライブ「FD-05」とインターフェイスカートリッジ。本製品は1DDと2DDに対応していた。これで8万9800円という価格だった。

左はデータレコーダで、右がQuick Disk。フロッピーディスクドライブがない人は、主にデータレコーダで保存していた。Quick Diskは、結果的にはディスクドライブにはかなわなかった。

キーボード
KEYBOARD

MSXではキーボードも統一されていた。したがって、キーの数も同じで、表記も基本的には共通だった。ここではMSXのフルキーとファンクションキーについて紹介しよう。

フルキー

MSXのキーは全72キーで構成されている(テンキーは除く)。並びは、英数字キーについてはJIS標準配列になっている。また、ひらがな/カタカナ入力については、JIS配列のほか、初期のMSXでは「あいうえおかきくけこ」という順番の50音配列を採用しているものも少なからずあった。むろん、パソコンに不慣れな子供やお年寄りでも、ひらがな/カタカナ入力がしやすいようにという配慮からである。

MSXには、PC/AT互換機には見られない独自のキーがいくつかあるので紹介しておくと、[かな(カナ)]は、ひらがな/カタカナ入力モードの切り替えをするためのキー。その後でフルキーを押すとそれぞれのキーが入力できた。[GRAPH]は、グラフィックスキャラクタを入力するためのキー。同時にフルキーを押すことで、罫線文字やトランプのマーク、「百千万時分秒年月火水木金土日」などの文字が入力できた。そして[STOP]は、一時停止のためのキー。BASICプログラムの実行やプログラムリストの表示を止めることができた。

ちなみに[STOP]キーは[CTRL]キーと同時押しすることで、実行されているBASICプログラムの中断や、表示中のプログラムリストの中断ができる。このキー操作が身に付いたときこそ真のMSXユーザーといえるだろう。

[SELECT]キーは、アプリケーションがときどき利用する程度だった。[HOME/CLS]は、通常は押すと文字カーソルが画面左上に移動するためのキーであるが、[SHIFT]キーと同時押しをすることで画面クリアができた。

残りの[ESC][TAB][CAPS][INS][DEL][BS]キーについてはPC/AT互換機とほぼ同じ役割だ。なお、[リターン]キー(実際には湾曲した矢印の図形が記されている)はPC/AT互換機の[Enter]キーに対応する。

ファンクションキー

MSXのキーボードの上部には[F1]~[F5]のファンクションキーが存在している。これらは[SHIFT]キーと併用することで[F6]~[F10]に切り替わる。

ファンクションキーには、BASIC環境の操作を短縮する「RUN」や「LIST」などの命令が設定されており、これを使うことで、BASIC環境での操作がワンタッチで行えた。ファンクションキーはカスタマイズができるため、自分なりに設定を変えていたユーザーも多かった。

ちなみに、画面下のファンクションキー表示はBASIC環境で「KEY OFF」を実行すれば消すことが可能だ(表示はKEY ON)。

キーの数は決まっていたが、材質などはメーカーの自由であった。そのため、安いMSXではゴム製の押しにくいキーもあった。逆に高級機では、キータッチがよいものが使われていた。

入力装置
INPUT DEVICE

MSXにはジョイスティックポートが標準で備わっており、ゲームには必須のものになっている。同じポートにはマウスやトラックボールなどを接続して使うことができた。

ジョイスティック

MSXには汎用のジョイスティックポートが搭載されている。MSX1では1ポート以上、MSX2以降は2ポートが標準仕様だ。コネクタはアタリ仕様(Pin8がGNDのため、厳密には擬似仕様)と呼ばれる台形の9ピン端子。ゲーム好きなMSXユーザーなら、少なからずここにゲームパッドやジョイスティックを接続して使っていたはずだ。

コナミなどから「グラディウス」などの本格的なシューティングゲームが発売されるようになると、シビアな操作が要求されるようになってきた。そのため、おもちゃ感覚のジョイスティックでは飽き足らないユーザーが増加。そこで登場したのが「ASCII STICK」(アスキー)や「PASOKO」(三和電子)といったハイスペックなジョイスティックだ。これらにはアーケードマシン用の部品がそのまま使用されており、操作感覚は抜群。また、ASCII STICKの後継機「ASCII STICK II TURBO」にはTURBOスイッチがあり、トリガーボタンのAまたはBについて、オート連射が可能。これで随分とゲーム攻略で助かったゲーマーも多かったはずである。

マウス

ジョイスティックはMSX1から接続できたが、マウス、トラックボール、タブレットなどの入力装置はMSX2からサポートされた。PC/AT互換機などでは、マウスはマウス端子に、タブレットはUSB接続などと、入力装置によって接続先が違ったりしているが、MSXの場合はいずれもジョイスティック端子に接続して利用できる。

ちなみに、マウスの移動距離などの情報はMSX-BASICのPAD(n)関数で得ることができた。MSXの場合、マウスポインタがスプライトで簡単に表現できることもあって、それ以降マウス対応のプログラムが急増することになる。MSX2ではパソコンによるお絵かきも盛んに行われたため、マウスを利用しているユーザーはかなり多かったと思われる。

やがて、MSXのグラフィックス能力が高くなると、HALNOTEやMSXViewなど、マウス必須のGUI環境がようやく登場することになった。

MSX用のいろいろな入力装置。左からトラックボール、操縦桿型のジョイスティック、MSXマウス、そしてアスキースティックだ。今も昔も入力装置に大きな変化はない感じだ。

インターフェイス
INTERFACE

MSXの標準インターフェイスは今に比べると少なかったといえる。具体的には、RF出力、ビデオ出力、アナログRGB出力、プリンタ、カセットインターフェイスなどがあった。

画面表示用端子

PC/AT互換機では、専用のディスプレイに接続するのが通例だが、ホームパソコンたるMSXでは家庭での利用という状況を考えて、3種類の画面表示用端子が用意されていた。

まずRF出力は、MSXの画面をテレビのアンテナ端子に接続して表示するもの。色がにじむので画質的はもう1つだが、テレビがあれば誰でも使えるというお手軽さから、もっとも利用されていた。

ビデオ出力は、NTSC信号（米国、日本など）を出力するもので、テレビなどのビデオ入力端子に接続して使う。音を出すには、同時に音声端子も接続する必要がある。画質はまあまあで、40桁表示ならばプログラミングも問題ない。しかし、80文字表示をするにはちょっとつらかった。

アナログRGB出力は、現在の専用ディスプレイと同様のもの。にじみはまったくなく80桁表示もクッキリとしていたため、プログラミングをしたり、よりキレイな画面でゲームをプレイしたい人が利用していた。アナログRGBに接続するには、21ピン端子に対応するディスプレイが必要だった。アナログ21ピン端子付きのテレビは、ほとんどなくなってしまった。しかし現在でもソニー製のテレビの一部機種に変換ケーブルを使用することでAVマルチ入力に接続可能である。

カセットインターフェイス

カセットインターフェイスは、FDDが普及するまでの間利用されていたものだ。ここにラジカセなどを接続して、カセットテープにプログラムの保存を行っていたのだ。データを音声信号に変換して磁気テープに記憶するというしくみである。ピーピーガーガと鳴る音に懐しさを覚える人も多いはず。記録／読み出し速度は痛烈な遅さで、涙なくしては語れないという感じだ。

カセットインターフェイスには入力／出力端子のほかにコントロール端子があり、MSXからモータのオン／オフが制御できた。専用のデータレコーダを接続しておけば、プログラムの保存／読み出し時には自動的にテープが回ってくれるため、ラジカセのようにタイミングを計るわずらわしさがなく、また頭出しもできたのでけっこう重宝された。

プリンタ

MSXが発売されていたころは、ドットインパクト方式や熱転写方式のプリンタが主流だった。

プリンタの使用用途は、プログラムリストのプリントアウトや、グラフィックスの出力、そしてワープロの印刷が主。

MSXのプリンタインターフェイスはセントロニクス仕様だったため、他機種用のプリンタを接続することもできたが、それではMSX特有のひらがなやグラフィック文字が印刷できないため、専用プリンタを接続するのが現実的だった。

MSXの背面はこんな感じ。暗くて見えにくいが、カセットインターフェイス、プリンタインターフェイス、RF端子、ビデオ／オーディオ端子、アナログRGB端子が並んでいる。

周辺装置
DEVICE

現在のPCと同様にMSXにもさまざまな周辺装置が発売されていた。その中から比較的よく使われていたもの。話題性が高いものなどをいくつか紹介してみよう。

通信モデム

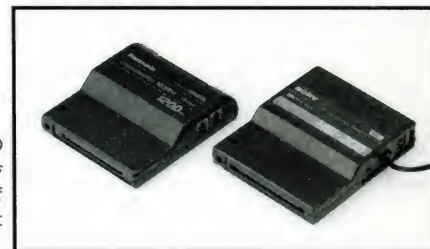
パソコン通信が流行っていた1986年あたりには、モデムカートリッジが普及した。しかし、当初は300bpsという速度。しかし、モデムカートリッジにはオートログイン機能やXmodemというプロトコルが搭載されるなど、機能的には便利なものだった。その後、すぐに1200bps対応したものが登場。しかし、外部モデムの機能向上の動きが激しく、その後はユーザーもRS-232Cカートリッジに外部モデムを接続して使うことになった。

パソコン通信といえば、THE LINKSというMSX専用のビジュアル通信ネットワークも忘れられない。ここでは専用のリンクスモデムが使われており、1200bpsで半二重という速度だった。

バージョンアップアダプタ

MSX2が出たときに、MSXユーザーは涙した。そんなユーザーのために作られたのが、NEOS（日本エレクトロニクス）が発売したバージョンアップアダプタだ。これをスロットに差すことによって、MSXがMSX2に変身するという夢のアダプタだった。MSX2が出てから1年以上先の発売となったため、すでにMSX2を買ってしまっていたユーザーも多かったが、一部のユーザーには喜ばれた。ちなみに、MSX2+やMSX turboR用のバージョンアップアダプタは発売されていない。

ビデオデジタイザー

MSX2+になり自然画モードが使えるようになったとき、ソニーから発売されたのが「HBI-V1」というビデオデジタイザーだ。これを使うと、SCREEN10～12の自然画だけでなく、SCREEN8という256色モードの画像をパソコンに取り込むことができた。現在のPC/AT互換機用のビデオキャプチャカードとは比べるべくもないが、それでも好きな画像がMSXにとりこめた喜びは大きかった。

1200bpsと300bpsのモデムカートリッジ。今となっては、この遅さでパソコン通信をしていたのがウソのよう。

■知っていると何気に便利

MSX-DOS講座

MSX-DOSはMSX用のOSで、64Kバイトのメモリ空間をすべて使ってファイル処理をしたり、プログラム開発を行うことができる。むろんアプリケーションの実行も可能だ。そして、本誌に収録したBASIC版MSXPLAYerには、MSX-DOSもまた収録されている。これは使ってやらねばなるまい。というわけでMSX-DOSの基本的な使い方を解説するぞ。

BASIC版
MSXPLAYerに収録

MSX-DOSの起動と終了

MSX-DOSを起動するには、MSXDOS.SYS、COMMAND.COMという2つのファイルが入ったディスクをAドライブに入れてから起動すればよい。

本誌に付属する「BASIC版MSXPLAYer」でMSX-DOSを使いたいときには、まずBASIC版MSXPLAYerを起動してから、仮想ディスクBを選択する。ここにMSXDOS.SYS、COMMAND.COMが入っているのだ。後はリセットすればMSX-DOSが起動する。もちろん、MSXDOS.SYS、COMMAND.COMを、仮想ディスクCやD、あるいは実ディスク（MSX実機でフォーマットしたFDD）にコピーしておけば、どのディスクからでも起動できるぞ。

画面に「A>」と表示されたのを確認してほしい。これは「プロンプト」と呼ばれる文字で、「現在コマンド（命令）の入力待ち状態ですよ」という意味だ。この状態でコマンドライン（コマンドを入力する行のこと）に対して実際にコマンドとなる文字列を入力して［リターン］キーを押すことで、何らかの処理をさせることができるわけだ。

A>DIR［リターン］

と入力してみよう。DIRとはDIRectry（ディレクトリ：名簿の意）の略。これでディスクに保存されているファイルの一覧が表示される。表示が終わると再び「A>」とプロンプトが表示され、コマンドの入力待ち状態になる。

このように、プロンプトに対してコマンドを入力し、プログラムが実行され、終わると再びプロンプトに戻る、という繰り返しがMSX-DOSによる操作の基本的な流れとなる。

次はMSX-DOSの終了だ。Windowsには「Windowsを終了する」という操作が存在するが、MSX-DOSには「終了する」という概念はない。したがってMSX-DOSを終了したいときにはMSXPLAYerをリセットするか、もしくはMSXPLAYerを通常どおりに終了すればオッケーだ。

BASIC版のMSXPLAYerを起動したら、仮想ディスクBを選択して［RESET］ボタンを押せばMSX-DOSが起動するのだ。

コマンドを使う

MSX-DOSでは、コンピュータへの命令を「コマンド」と呼んでいる。コマンドを入力すると、それに対応するプログラムが動作して、処理が始まるというわけだ。

MSX-DOSには、標準で使えるコマンド（内部コマンドと言う）がいくつか用意されている。そのすべてを覚える必要はないけれど、とりあえず、BASIC、DIR、COPY、DEL、REN、TYPEコマンドについてだけ覚えておくといいだろう。ちなみにBASICは、MSX-BASIC環境に移るためのコマンド。DIRは、ファイルの一覧を表示するためのコマンド。TYPEは、テキストファイルの内容を表示するためのコマンド。そしてCOPY、DEL、RENはファイルのコピー、削除、名前の付け替えのためのコマンドだ。使い方は p.173の表をご覧になっていただきたい。

コマンド入力時の注意

●コマンド文字は正しく

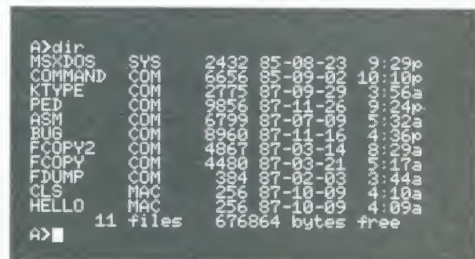
コマンドのスペルは間違えないようにしよう。「DIR」を「DLR」として実行すると「Bad command or file name（コマンドまたはファイル名が間違っています）」というエラーになってしまう。コマンドはたった1文字間違っていても働かないので［リターン］キーを押す前には要チェック。

●半角スペースを入れる

ファイル名などの引数（パラメータ）を入力する場合には、コマンド名と引数の間や、引数と引数の間に1文字以上の半角スペースを入れよう。「DEL FILE.TXT」とすべきところを、半角スペースを入れ忘れて「DELFILE.TXT」とすると、MSX-DOSは「DELFILE.TXT」というコマンドが入力されたと勘違いしてしまうのだ。逆に1文字以上の半角スペースを入れることについては問題ない。もっとも余計に入れても無意味だが。

●大文字小文字は区別なし

コマンドの大文字と小文字は区別しなくても大丈夫だ。「DIR」を「Dir」と入力してもなんら問題はない。MSX-DOSは大文字と小文字を区別しない仕様になっているのだ。

DIRコマンドを実行してみたところ。ファイルの一覧が表示される。表示が終わると、再びプロンプトが表示されるのだ。

●表 MSX-DOSの基本コマンド

	コマンド	使い方	意味	例
BASIC	BASIC <ファイル名>	BASIC環境に移るためのコマンド。	<ファイル名>を指定するとBASICに移行した後で<ファイル名>のBASICプログラムが実行される。	●A>BASIC PROG.BAS BASICに移行してPROG.BASを実行する
COPY	COPY <コピー元ファイル名> <コピー先ファイル名>	<コピー元ファイル名>を<コピー先ファイル名>にコピーするためのコマンド。	たとえば「COPY FILE1.TXT FILE2.TXT」として実行すればFILE1.TXTがFILE2.TXTにコピーされる。ワイルドカードを使うと一度に複数のファイルをコピーできる。コピー先として複数のファイルを指定して+で接続することでファイルの連結を行うこともできる。	●A>COPY FILE.TXT WORK.TXT FILE.TXTをWORK.TXTという名前でコピーする ●A>COPY *.DOC *.TXT 拡張子が「.DOC」のファイルを、拡張子を「.TXT」にしてコピーする ●A>COPY *.* B: すべてのファイルをBドライブにコピーする。このようにコピー先のファイル名を明記しないと同じ名前でコピーされる ●A>COPY FILE1.TXT+FILE2.TXT OUTPUT.TXT FILE1.TXTとFILE2.TXTを連結してOUTPUT.TXTを作る
DATE	DATE [<年>-<月>-<日>]	日付の表示/変更を行うためのコマンド。	「DATE」と実行すると日付が表示される。実機ではMSXの内部時計が参照されたが、MSXPLAYerではWindowsの日付が参照される。実機では<年>-<月>-<日>を指定して「DATE 2002-1-1」のようにすることで内部時計を再設定できるが、MSXPLAYer版では無効になる。	
DEL	DEL <ファイル名>	<ファイル名>で指定したファイルを削除するためのコマンド。	たとえばFILE.TXTを削除するなら「DEL FILE.TXT」と実行すればよい。ワイルドカードを使うと一度に複数のファイルを削除できる。このときには、「Are you sure? (Y/N)」と確認メッセージが表示されるので削除してよければ[Y]キーを押そう。	●A>DEL FILE.TXT FILE.TXTを削除する ●A>DEL *.TXT 拡張子が「.TXT」のファイルをすべて削除する
DIR	DIR [<ドライブ名>] [/W] [/P]	ディスクに記録されているファイルの一覧を表示するためのコマンド。	「DIR /W」とするとワイド表示になり、1行に複数のファイル情報が表示される。「DIR /P」とするとページ表示になり、1画面以上のファイル情報が表示される場合には「Strike a key when ready」と表示され、何かのキー入力があるまで一時停止する。ワイルドカードを使うと特定のファイルだけを表示できる。	●A>DIR ファイルの一覧を表示する ●A>DIR /W ファイルの一覧をワイド表示する ●A>DIR *.TXT 拡張子が「.TXT」のファイルだけ表示する
FORMAT	FORMAT	ディスクのフォーマット(初期化)を行うためのコマンド。	MSXPLAYerでは無効となっている。	
MODE	MODE <数値>	1行の文字表示数を<数値>に切り替えるためのコマンド。	数値の範囲によって画面モードも切り替わる。1~32を指定したときにはSCREEN1、33~40を指定したときにはSCREEN0、41~80を指定したときにはSCREEN0の80桁モードとなる。	●A>MODE 40 テキストの1行の文字数を40桁に切り替える。SCREEN0の40桁モードになる ●A>MODE 80 テキストの1行の文字数を80桁に切り替える。SCREEN0の80桁モードになる
PAUSE	PAUSE <コメント>	バッチ処理用のコマンド。	一時的にバッチ処理を停止したい場所に記述する。バッチ処理中に本コマンドに出会うと画面に「Strike a key when ready」と表示される。何かのキーを押されると続きが実行する。	
REM	REM <コメント>	バッチ処理用のコマンド。	コメントを書くのに使う。処理に影響はない。	
REN	REN <旧ファイル名> <新ファイル名>	<旧ファイル名>を<新ファイル名>に変更するためのコマンド。	たとえば「REN FILE.TXT WORK.TXT」とすると、FILE.TXTというファイル名がWORK.TXTに変更される。ワイルドカードを使うと一度に複数のファイル名を変えることができる。	●A>REN FILE.TXT WORK.TXT 「FILE.TXT」を「WORK.TXT」に変更する ●A>REN *.TXT *.DOCァイ 「.TXT」という拡張子のファイルについてすべての拡張子を「.DOC」に変更する
TIME	TIME [<時>:<分>:<秒>]	時間の表示/変更を行うためのコマンド。	「TIME」を実行すると時間が表示される。実機ではMSXの内部時計が参照されたが、MSXPLAYerではWindowsの時間が参照される。実機では<時>:<分>:<秒>を指定して「TIME 02:30:00」のようにすることで内部時計を再設定できるが、MSXPLAYer版では無効になる。	
TYPE	TYPE <ファイル名>	<ファイル名>で指定したテキストファイルの内容を表示するためのコマンド。	テキストファイル以外を対象にすると表示が崩れるので注意。また、テキストファイルでも日本語のテキストファイルは表示できない。	●A>TYPE FILE.TXT 「FILE.TXT」の内容を表示する
VERIFY	VERIFY {ON|OFF}	ディスクへの書き込みの確認をするかを指示するためのコマンド。	「VERYFY ON」を実行しておくと、一度ファイルを書き込んだ後で、ファイルが読み出され、比較する。	

注: [...]は省略可能であることを意味する。{A|B}はAとBのいずれかを指定することを意味する。<...>はファイル名などの引数。

テンプレート

コマンドを入力する際に、ミスをすることもよくある話だ。たとえば、

A>COPI FILE.TXT OUTPUT.TXT [リターン]

とすると「COPY」と入力されるべきところが「COPI」となっているためにエラーとなってしまう。そこで再度入力するわけだが、同様なミスを連続して起こさないとも限らない。これではコマンドを入力がイヤになって当たり前というものだろう。

そこで紹介したいのが「テンプレート機能」。実は、1回前にコマンド入力した文字列は「テンプレート」という領域に保存されている。これを適宜呼び出すことで、コマンド入力の手間を軽減することができるのだ。

たとえば上記のコマンドを入力すると、エラー表示後に再びプロンプトが表示される。ここで[→]キーを3回押す。すると「COP」のように前回入力した文字が順次コマンドラインに表示されていく。次に[Y]キーを押す。これで「COPY」となる。最後に[↓]キーを押す。これで残りの文字列がすべて表示され、ついにコマンドラインには「COPY FILE.TXT OUTPUT.TXT」と正しいコマンドが完成する。あとは[リターン]キーを押せばいいわけだ。

テンプレートに記憶されている内容は直接ユーザーに見えないため、慣れないと少し不安だ。しかし、入力ミスしたコマンドの修正や、連続して同じようなコマンドを入力するときには、テンプレート機能が欠かせない。右表のすべてを覚える必要はないけれど、とりあえず[→]キーを押すと前回のコマンドから1文字ずつ取り出されることと、[↓]キーを押すと前回のコマンドがまるごと表示される、ということぐらいは知っておきたいところだ。

●図 テンプレート機能

●表 テンプレート機能

キー	意味
[↓]	テンプレートからコマンドラインにすべてコピーする
[↑]	コマンドラインの文字列をすべて消す
[→]	テンプレートからコマンドラインに1文字ずつコピーする
[←]	コマンドラインの最後の1文字を消す
[SELECT]+○	テンプレートの○で指定した文字までコマンドラインにコピーする
[INS]	挿入モードにする。挿入モード中に文字入力してもテンプレートの参照位置は変わらない
[DEL]	テンプレートの参照位置を1文字分進める
[HOME]	コマンドラインの文字列をテンプレートに記憶する

コントロールキー

DIRやTYPEコマンドを実行すると画面に文字がずらずらと表示される。このとき表示される文字が1画面分以上あると、画面がスクロールしてしまって、じっくり表示内容を確認することができない。こうしたときには途中で[CTRL]+[S]キーを押してみよう。これで表示が一時停止してくれる。続きが見たいときには何かのキーを押せばよい。

また、途中でコマンド実行を終了したいときもある。その場合には[CTRL]+[C]キーを押す。これでコマンドの実行が中断する。ただし、これらのコントロールキーによる操作は、プログラムによっては効果がないので、そのときは素直にあきらめること。

また、後で解説するバッチコマンド処理中に、[CTRL]+[C]キーを押すと、「Terminate batch file(Y/N)」と表示される。[Y]キーを押せばバッチ処理を中断することもできるのだ。

コマンドを中断したいときには[CTRL]+[C]キーを押してみよう。押したところで「^C」という文字が表示され、すぐにプログラムが終了。「>A」というプロンプトが表示されるのだ。

ワイルドカード

ファイルを指定するときにワイルドカード文字("*"と"?")が使える。これは、トランプのジョーカーのような役割をする文字で、一度に複数のファイルを処理の対象にできるというもの。"*"は任意の文字列として機能し、"?"は任意の1文字として機能する。たとえば拡張子が「.TXT」のファイルだけを表示したいときには、

A>DIR *.TXT [リターン]

とすればよい。これで「FILE.TXT」や「TEXT.TXT」が表示される。

同様に、

A>DIR ?.TXT [リターン]

とすると「F.TXT」や「A.TXT」のようにベースネームが1文字のファイルだけが表示されるのだ。

●表 ワイルドカードの例

例	意味
DIR *.TXT	F.TXT、FILE.TXTなど、ベースネームは何でもよく、拡張子が".TXT"のファイルを一覧表示する
DIR FILE.*	FILE.TXT、FILE.DOCなど、ベースネームが"FILE"のファイルを一覧表示する
DIR FILE?.TXT	FILE1.TXT、FILES.TXTなど、ベースネームが"FILE"で始まる5文字で、拡張子が".TXT"のファイルを一覧表示する
DIR ???.TXT	ABC.TXT、123.TXTなど、ベースネームが3文字で、拡張子が".TXT"のファイルを一覧表示する
DIR F??E.TXT	FILE.TXT、FINE.TXTなど、ベースネームが"F"で始まり"E"で終わる4文字で、拡張子が".TXT"のファイルを一覧表示する
DIR *.*	すべてのファイルを一覧表示する
DIR ?*.?*	このような使い方をしても意味はない(*.*と同じ)

注:ベースネームとは、ファイル名のうち拡張子を除く左の部分を言う。

仮想ドライブ

MSX-DOSには、1台のドライブをAドライブとしてだけではなく、Bドライブとしても使うことが可能な「仮想ドライブ」という機能がある。これは、1台のフロッピーディスクドライブを有効に使うために備わっている機能である。MSXPLAYerもフロッピーディスクドライブが1台接続されている仕様となっているため、Bドライブを指定すると自動的に仮想ドライブ機能が働くことになる。

たとえば、

A>DIR B: [リターン]

とすると「Insert diskette for drive B: and strike a key when ready」(Bドライブにディスクを入れて何かのキーを押してください)と表示される。ユーザーがAドライブに入っているディスクを抜いて、別のディスクに入れ替える。何かのキーを押すとディスクはBドライブに入っているものとして機能する、という具合である。

現在、ディスクがAドライブになっているか、あるいはBドライブになっているかは、上記のメッセージによって判断する必要がある。万一、Bドライブになっているとき、Aドライブのディスクを入れてしまうと、別のディスクの内容が表示されることになるので注意しよう。

なお、Aドライブに戻るときには「Insert diskette for drive A: and strike a key when ready」とメッセージが表示されるので、指示に従ってAドライブに入っているべきディスクを正しく選択すること。

MSXPLAYerのディスク切り替え

MSXPLAYerの画面下には、A、B、C、D、FDDというボタンが並んでいるため、一見MSXにフロッピーディスクドライブが5台接続されているかのように見えるかもしれない。しかし、これはあくまでもディスクのメディアを意味している。MSXPLAYerからアクセスできるのはAドライブのみ。MSXPLAYerの下のボタンは、このAドライブに、A、B、C、D、FDDのどの仮想ディスクを入れて使うかを選択をするためのものだ。混乱しないようにしておこう。

MSXPLAYerの下にあるディスクは、ドライブではなくて、あくまでもディスクのメディアなのだ。選んだ仮想ディスクがAドライブに入り、使えるようになるのだ。

カレントドライブ

カレントドライブとは、現在処理対象となっているドライブのことだ。たとえば、

A>DIR [リターン]

として実行したときには、カレントドライブの内容が一覧表示される。カレントドライブがAであれば、このコマンドを実行するとAドライブの内容が一覧表示されるのだ。カレントドライブが暗黙的に対象ドライブを示してくれているおかげで、いちいちドライブ名を入力する手間が省けるようになっているわけ。ちなみに、ドライブ名は「<ドライブ名>+":"」で表現する。したがって、この機能がないとしたら、いちいち、

A>DIR A:
A>COPY A:FILE.TXT A:WORK.TXT

とドライブ名(A:)を入力しなければならなかったはずだ。

現在カレントドライブがどこを示しているかはプロンプトを見ればわかる。「A>」と表示されていればカレントドライブはA。「B>」となっていればカレントドライブはBである。

カレントドライブをAからBに切り替えるには、

A>B: [リターン]

とする。以後プロンプトが「B>」となり、処理対象がBドライブになる。BからAに戻す場合には同様にして、

B>A: [リターン]

とする。

DIR、COPY、DEL、REN、TYPEコマンドを使うとき、ドライブ名を指定しなければ、いずれもカレントドライブが対象になる。しかし、あえてドライブ名を指定することもできる。たとえばBドライブのファイルを一覧表示したいときには、

A>DIR B: [リターン]

とすればよい。また、Bドライブのテキストを表示したければ、

A>TYPE B:README.TXT

のようにする。

なお、Bドライブを指定したときには、MSXPLAYerの場合、仮想ドライブ機能が働くことをお忘れなく。

プログラムの実行

MSX-DOSは、DIRやCOPYなどの基本コマンドだけ使っていても、おもしろくもなんともない。通常はどこからか(たとえばインターネットなどから)、開発ツール、ユーティリティ、アプリケーション、あるいはゲームをダウンロードして実行するわけだ。

最近では、MSXを使ってプログラムを入手するより、Windows環境で入手することの方がほとんどだろうから、これを利用する手順を示しておこう。

①2DDフォーマットしたフロッピーディスクを用意して、Windowsのフロッピーディスクドライブに入れる
②ダウンロードしたMSX-DOS用の実行ファイル一式を、②で用意したフロッピーディスクにWindows環境でコピーする(もしファイルが圧縮されていればWindows環境で解凍しておこう)
③MSXPLAYerの仮想ディスクを「FDD」に切り替える
④MSX-DOSでコマンド名を入力して実行する

①〜③でMSXPLAYerがアクセスできる場所に実行ファイルを移動し、④でコマンドを実行しているわけだ。④については、実行ファイルがPROG.COMというファイル名だった場合には、

A>PROG [リターン]

とすればよい。

MSX-DOS用のプログラムの実行ファイルは拡張子が.COMだが、それ以外に必要なファイルがないとは言い切れない。また、コマンドを実行する際には引数が必要となることもある。したがって、実行する前には付属の解説などをよく読んでおくことをお忘れなく。

実機のMSXで使っていたフロッピーディスクに保存されているMSX-DOSのプログラムを実行したいときには、それをフロッピーディスクドライブに入れて、MSXPLAYerの仮想ディスクを[FDD]に切り替えて実行するだけだ。

なお、MSXのフロッピーディスクを使うときには、本誌TIPSにもあるように、書き込み禁止状態にしてから行っていただきたい。

バッチコマンド

MSX-DOSにはバッチコマンドという機能があって、ユーザーがその場で新しいコマンドを作ることができる。そして、これを利用すると、あるコマンドを別の名前で実行したり、複数のコマンドを連続して実行させたりが、いとも簡単にできてしまう。たとえば、

```
A>MODE 80
```

を実行するのに、いちいち7文字も入力するのは面倒だと感じたときには、まず、

```
MODE 80
```

という内容のW.BATというファイルを作成する。これであとは、

```
A>W [リターン]
```

とすればW.BATが読み込まれ「MODE 80」が実行される(画面が80桁モードになる)。

このように、テキストファイルを作り、その中に実行したいコマンドを記述する。あとは拡張子を.「BAT」としておけば、以後コマンドとして機能するのがバッチコマンドというものなのだ。

バッチファイル内には複数のコマンドを羅列することもできる。バッチとは束という意味があり、本来はコマンドを束ねて使うためのものなのだ。たとえば、

```
MODE 80
DIR
```

という内容のW.BATというバッチファイルを作成し、先ほどと同様にして実行すると、最初に「MODE 80」を実行され、続けて「DIR」が実行される。コマンドはいくつでも記述できる。MSX-DOSではバッチコマンドを利用することで、単機能のコマンドを組み合わせて複雑な処理を行わせることができるようになっているのだ。

また、少し難しい話になるが、バッチコマンドには引数を渡すことができる。たとえば、先ほどのバッチファイルの内容を、

```
MODE %1
DIR %2
```

のように変更する。これを実行するときには、

```
A>W 50 *.TXT
```

のようにする。すると、1つめの引数「50」が%1と、2つめの引数「*.TXT」が%2と置き換わり、結果的には、

```
MODE 50
DIR *.TXT
```

として実行されることになる。これはバッチコマンドの入力時の指定に柔軟性を持たせたいときに役立つ機能だ。

なお、バッチファイルを作る場合には、下で解説するリダイレクトを用いてもいいが、MSX-DOS用のテキストエディタのほか、MSXPLAYerならばWindows用のテキストエディタ(メモ帳など)を使って作成してもオッケーだぞ。

AUTOEXEC.BAT

MSX-DOSが起動したとき、起動ディスクにAUTOEXEC.BATというバッチファイルが含まれていると、自動的にこれを実行してくれる。たとえば、起動後そのままBASICに移りたいのなら、

```
BASIC
```

という内容のAUTOEXEC.BATを作っておけばよい。

また、BASICに移ったと同時にプログラムを実行したければ、

```
BASIC AUTOEXEC.BAS
```

のようにBASICのファイル名を記述する。

ちなみにBASICから再度MSX-DOSに戻るには

```
call system
```

とすればよい。

デバイスファイル

MSX-DOSには、画面、キーボード、プリンタなどについても一般のファイルと同じように扱えるしくみが備わっている。それがデバイスファイルだ。ファイルというからには名前があり、画面やキーボードにはCON、プリンタにはPRNといった名前が付いている。そしてたとえば、

```
A>COPY A.TXT PRN [リターン]
```

とすればA.TXTの内容をプリンタにコピー(印刷)できる。実際には、MSXPLAYerはプリンタをサポートしていないのでこれは使えないが、次にMSXPLAYerでも使える便利技を2つ紹介しておこう。

複数のテキストファイルを一度に表示する

テキストファイルを画面表示するには普通TYPEコマンドを使う。しかし、TYPEコマンドではワイルドカードが使えない。したがって、複数のテキストファイルを表示するときには、TYPEコマンドを何度か実行する必要がある。このようなときには、

```
A>COPY *.TXT CON [リターン]
```

としてみよう。これはテキストファイルを画面にコピーせよ、という意味。COPYコマンドはワイルドカードが使えるので、一度に表示できるってわけだ。

エディタなしでテキストファイルを作る

次はテキストエディタがなくてもテキストファイルを作る方法。たとえばOUTPUT.TXTというテキストファイルを作りたいなら、

```
A>COPY CON OUTPUT.TXT [リターン]
```

とする。文字カーソルが次行先頭に移動したら、文字をどんどん入力していけばよい。

[リターン]キーを押すと改行することも可能だ。ただし、その場合には文字カーソルが次行に移ってしまい、前の行の編集はできなくなってしまうから注意してほしい。[リターン]キーを押す前であればテンプレート機能(p.174参照)を使って修正することができる。

すべての文字入力が終わったら[リターン]キーを押し、文字カーソルを行頭に移動。それから最後のおまじないとして[CTRL]+[Z]キーを押す。画面に「^Z」を表示されるので[リターン]キーを押す。これで入力した文字がOUTPUT.TXTに保存される。

もうおわかりだろうが、これもキーボードからファイルにデータをコピーするという意味だ。

```
A>copy con output.txt
mode 80
dir
^Z
        1 file copied
A>type output.txt
mode 80
dir

A>
```

エディタがなくても、このように文字を入力し、最後に^Zを入力すれば、その場でテキストファイルが作成できてしまうのだ。

他のディスクにファイルをコピーする

MSXPLAYerには仮想ディスクがA、B、C、D、FDDの5つ存在しているが、この仮想ディスク同士でファイルのやり取りをするには、MSX-DOSのCOPYコマンドと仮想ドライブを利用するのが手っ取り早い。

たとえば、仮想ディスクBに記録されているファイルをすべて仮想ディスクCにコピーしたいときには、まずMSXPLAYerで仮想ディスクBを選択。そして、

A>COPY A:*.* B: [リターン]

を実行する。これで、<ドライブA>に入っている仮想ディスクBの中のファイルを、<ドライブB>に入っている仮想ディスクCにコピーできる。

実行すると、仮想ディスクBのファイルがメモリに読み出される。このとき、メモリが一杯になるまで複数のファイルが読み込まれるので、ファイルを1つずつコピーしなくて済むのである。

しばらくすると、「Insert diskette for drive B: and strike a key when ready」と表示される。ここで、MSXPLAYerで仮想ディスクCを選択し、何かのキーを押す。今度はメモリ上のファイルが仮想ディスクCに書き込まれる。

終了すると、今度は「Insert diskette for drive A: ～」と表示される。MSXPLAYerで仮想ディスクBを選択し、何かのキーを押す……この繰り返しでファイルがコピーできるのだ。

同じ要領で、「FDD」(実ディスク)の内容を仮想ディスクCにコピーしたり、ディスクDの内容を「FDD」にコピーしたりも可能になる。

日本語MSX-DOS2

MSX-DOSが登場した後、アスキーから「日本語MSX-DOS2」が発売されている。こちらは日本語環境、階層化ディレクトリ、大容量メモリ、MS-DOS ver2.xx互換のファイルシステムなどがサポートされている。8ビットOSとしては最高の環境といえるだろう。いずれMSXPLAYerに日本語MSX-DOS2が搭載されることを期待しよう。

COPYコマンドで<Aドライブ>から<Bドライブ>にファイルをコピーしているところ。実際には、仮想ディスクBの内容をディスクDにコピーしている。今回はファイル数が少なかったので、一度の仮想ディスクの入れ替えでコピーが済んでしまったが、ファイル数が多い場合には、何度かディスクの入れ替えが指示される。

コマンドを作る

MSX-DOS用のプログラムは、もっぱらC言語などで作成するのが一般的だが、BASIC版MSXPLAYerには『MSX-DOSスーパーハンドブック』(BITS/アスキー刊)に掲載されていたMSX-DOS版のアセンブラを収録しておいた。また、サンプルプログラムとして、画面をクリアするCLSコマンドのソースと、"HELLO!"とメッセージを表示するHELLOコマンドのソースも収録してある。

それでは、MSX-DOSによるプログラミングへのアプローチとして、CLSコマンドのソースコードをアセンブルして、実際にCLSコマンドを作ってみよう。ちょうどMSX-DOSには画面をクリアするコマンドがないので、CLSコマンドがあると何気に便利だぞ。

CLSコマンドのソースがCLS.MACだ。これをアセンブルするには、

A>ASM CLS.MAC [リターン]

とすればよい。すると画面にアセンブルした結果が表示される。終了するとCLS.COMというファイルが作成されているはずだ。あとは、このまま、

A>CLS [リターン]

とすれば、画面に表示されている文字がすべて消えて新しくプロンプトが画面上に表示される。

CLS.MACではコントロールコードと呼ばれる特別な文字を画面に出力することで画面をクリアしている。アセンブラによるプログラム開発を習得するには、Z80とMSX-DOSのお勉強が必須だ。

●リスト CLS.MAC

```
;CLS
        ORG     00100H

COUT1   EQU     002H    ;OUTPUT 1 CHAR
CLS     EQU     00CH    ;CTRL-CODE
SYSTEM  EQU     00005H

        LD      E,CLS
        LD      C,COUT1
        CALL    SYSTEM
        RET

        END
```

CLS.MACをアセンブルしているところ。表示されているメッセージの1行が長いので80桁モードで実行している。表示されているのはソースコードと、生成されたマシン語だ。

MSXPLAYer向け

MSX-BASIC資料編

文法リファレンス

1. BASICの種類

MSXに標準で搭載されるBASIC言語は、MSX規格の拡張とともにいくつかのバージョンが作られています(下表)。

名称	内容	備考
MSX BASIC ver.1.0	MSX1のBASIC	
MSX BASIC ver.2.0	MSX2のBASIC	
MSX BASIC ver.3.0	MSX2+のBASIC	
MSX BASIC ver.4.0	MSXturboRのBASIC	
MSX Disk BASIC ver.1.0	MSX-DOS1下のDisk BASIC	
MSX Disk BASIC ver.2.0	MSX-DOS2下のDisk BASIC	MSXturboRで標準搭載

2. BASICの動作モード

BASICの動作モードには、入力された文(命令)を直接実行するダイレクトモードと、あらかじめ入力しておいたいくつかの文を順に実行するプログラムモードがあります。また、文を直接実行したり、プログラムを入力したりするときの状態を「コマンドレベル」と呼び、実行モードであるダイレクトモードやプログラムモードと区別します。

2-1　ダイレクトモード

コマンドレベルでBASICの文を入力し、[RETURN]キーを押すと、その文をすぐに実行するモードです。ダイレクトモードで入力した文は、メモリに記憶されません。

2-2　プログラムモード

コマンドレベルで文の前に番号を付けて入力し、[RETURN]キーを押すと、その文は実行されず、番号付きでメモリに記憶されます。この番号を「行番号」と呼び、行番号付きの文を「行」、行の集まりを「プログラム」と呼びます。プログラムはRUN文などで実行することができ、これを「プログラムモードでの実行」と呼びます。

● プログラムの作成

プログラムは、行の集まりです。行は、1つ以上の文を含む255文字以内の文字列で、行番号の順番に実行されます。1つの行に複数の文を含める場合は、文をコロン(:)で区切ります。行番号は0～65529の整数で、実行順序を表す他、GOTO文やGOSUB文のジャンプ先として使用します。文には大文字、小文字の区別がなく、小文字で入力してもすべて大文字に変換されて記憶されます。ただし、引用符(")で囲まれた文字や、REM文やアポストロフィー(')に続くコメント文は変換されず、そのまま記憶されます。

● プログラムの修正

プログラムの修正は行に対して行います。行える修正は次の通りです。

- **行の追加** ―― 追加する文に行番号を付けて入力する。入力した行は、行番号順に並び替えられて記憶される。
- **行の削除** ―― 削除する行の行番号のみを入力して[RETURN]キーを押す。複数の行をまとめて削除するには、DELETE文を使用する。
- **行の変更** ―― 変更する行の行番号に変更する文を付けて入力すると、入力した文が古い文に置き換わる(行の置き換え)。行の一部を修正するには、LIST文で修正する行を画面に表示し、直接書き換えて[RETURN]キーを押す。
- **行番号の振りなおし** ―― RENUM文を使って、行番号を振り直すことができる。

3. BASICの文法

BASICで使用できる文字、データ型、定数と変数、式と演算について説明します。

3-1　使用できる文字

BASICでは、英文字(大文字・小文字)、数字、片仮名、平仮名、記号、グラフィック文字を使用できます。文字については、キャラクタコード表も参照してください。

3-2　データ型

BASICで使用できるデータ型には、大きく分けて数値型と文字型があり、数値型は更に整数型、単精度実数型、倍精度実数型に分類できます(**図1-1参照**)。

● 文字型

文字型は、255文字以内の文字を連ねたデータ(文字列)です。BASICで使用できる文字すべてを使用できます。文字型のデータに許される演算は、文字列演算だけです。

● 整数型

整数型は-32768～+32767の範囲を持つ整数を表します。整数型の演算は、実数型の演算よりも高速です。そのため、実数型の精度を必要としない数値演算を行う場合は、整数型を使用すると高速なプログラムを作成できます。

● 単精度実数型

有効数字6桁の精度を持つ実数です。仮数部は-9.99999～9.99999の、指数部は-64～+62の範囲を持ちます。指数部を持つ実数型定数

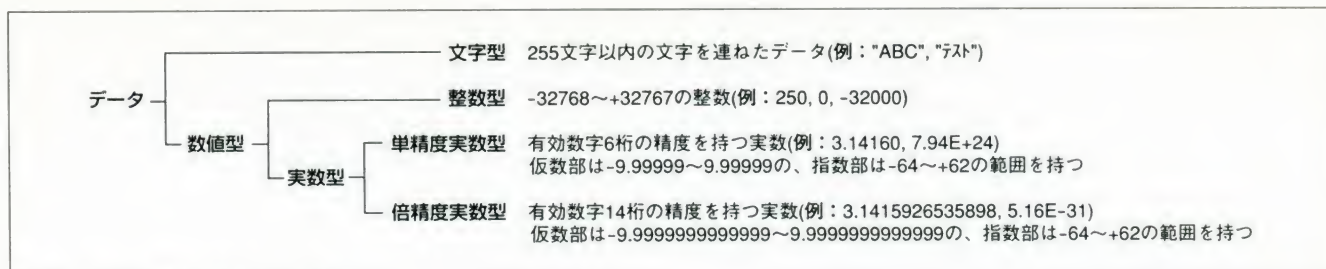



図1-1　BASICで使用できるデータ型

は、(仮数部)E(指数部)の形式で表現します。これは、(仮数部)×$10^{(指数部)}$を意味します(例：9.94E+24=9.94×10^{24})。

● 倍精度実数型

有効数字14桁の精度を持つ実数です。仮数部は-9.9999999999999～9.9999999999999の、指数部は-64～+62の範囲を持ちます。

3-3　型変換

数値型(整数型・単精度実数型・倍精度実数型)は、必要に応じて自動的に他の数値型に変換されます(暗黙的な型変換)。型の変換は、次の規則に従って行われます。

- 数値型の値を他の数値型の変数に代入する場合、値は代入先の型に変換される。
- 精度の異なる数値型の値を使った演算では、精度の低い方が高い方の型へ変換されて演算が行われる。
- 論理演算では、すべての値が整数型に変換され、結果も整数型の値になる。
- 実数型の値が整数型に変換される場合、小数点以下は切り捨てられ、切り捨てた結果が整数型の範囲(-32768～+32767)に収まらなければエラー(Overflow)となる。
- 倍精度型の値が単精度型に変換される場合、7桁目以降は四捨五入され、有効数字6桁になる。
- 15桁以上の値が倍精度型に変換される場合、15桁目以降は四捨五入され、有効数字14桁になる。

数値型と文字型の暗黙的な型変換は存在しません。明示的な変換を行うには、VALやSTR$を使用します。

3-4　定　数

定数は書き換えることのできない値を持つデータのことです。定数の表現方法はデータ型によって異なります。

● 文字型

文字定数は、255文字以内の文字列を引用符(")で囲んで表現します。数字を引用符で囲むと文字列とみなされます(例： "Hello, world.", "3.1415926535898")。

● 整数型

整数型定数には、基数によって4つの表現方法が存在します。

- **10進形式** ―― 0～9の数字を並べた-32768～+32767の整数。負の値の場合、マイナス符号(-)が必要だが、正の値の場合、プラス符号(+)は省略できる。また、-32768～+32767に収まる実数定数の直後に%を付けると、変換規則に従って整数型定数となる(例：-142, +10, 32459, 3.14%)。
- **2進形式** ―― &Bの直後に0～1の数字を並べた形式。範囲は&B0～&B1111111111111111。&B1111111111111111～&B1000000000000000は、-1～-32768に対応(例：&B10110110)。
- **8進形式** ―― &Oの直後に0～7の数字を並べた形式。範囲は&O0～&O177777。&O177777～&O100000は、-1～-32768に対応(例：&O4367)。
- **16進形式** ―― &Hの直後に0～9とA～Fの16進数字(A～Fは10～15に対応)を並べた形式。範囲は&H0～&HFFFF。&HFFFF～&H8000は、-1～-32768に対応(例：&HF380)。

● 単精度実数型

6桁以内の実数です。Eを使った指数表記も可能です(仮数部は-9.99999～9.99999、指数部は-64～+62)。

負の値の場合、マイナス符号(-)が必要ですが、正の値の場合、プラス符号(+)は省略可能です。また、7桁以上の実数定数の直後に!を付けると、変換規則に従って単精度型定数となります(例：-142.156, +1.12358E17, 3E-56, 3.1415926535898!)。

● 倍精度実数型

14桁以内の実数です。Eを使った指数表記も可能です(仮数部は-9.9999999999999～9.9999999999999、指数部は-64～+62)。Eの代わりにDを使うこともできます。

負の値の場合、マイナス符号(-)が必要ですが、正の値の場合、プラス符号(+)は省略可能です。また、15桁以上の実数定数の直後に#を付けると、変換規則に従って倍精度型定数となります(例：-15492.711825396, +1.12358E17E-56, +3E56, 3.141592653589845635#)。

3-5　変　数

変数はプログラム中のデータを格納するもので、格納場所としてメモリが使用されます。変数には英数字1～2文字の名前(変数名)を付けて、他の変数と区別します。変数に値を代入する場合、変数の型と代入する値の型が一致しているか変換可能である必要があります。

代入を行う前は、数値型変数は0に、文字型変数はヌルストリング("")に初期化されています。

● 変数名

変数名に使用できる名前は、次の規則に従っている必要があります。

- 英数字1～2文字であること。
- 先頭1文字は、英字(A～Z)であること。
- 2文字目は英字(A～Z)か数字(0～9)であること。
- 変数名が予約語を含まないこと。

また、3文字以上の変数名を指定すると、2文字目までしか変数名として扱われません。

● 変数の型

変数の型は、変数名の直後に型宣言文字を置くことによって指定できます。型宣言文字は次の4つです。

型宣言文字	型
%	整数型
!	単精度実数型
#	倍精度実数型
$	文字型

型宣言文字を付けると、変数名が同じでも、異なる変数として扱うことができます(例：A#, A$は別の変数として扱われる)。

型宣言文字を省略すると、通常は倍精度実数型とみなされますが、DEFINT／DEFSNG／DEFDBL／DEFSTRによる型宣言が行われている場合は、宣言された型とみなされます。

● 配列変数

配列変数は、同じ型のデータを順に並べた構造を持つ集合です。先頭のデータから順に、0から始まる整数が割り振られており、この整数を使って各要素にアクセスします。整数は、変数名か型宣言文字の直後に括弧か角括弧で囲んで指定します(例：A(0), B$[10])。

配列変数を使用するには、DIM文で変数名と型、添え字の最大値を宣言します。DIM文を実行せずに配列変数を使うこともできますが、その場合、添え字の最大値は自動的に10に設定されます。

複数の添え字を持つ配列変数を使うこともできます。添え字の数が配列変数の次元になります。

● 変数の使用するメモリ

変数の使用するメモリの量は、変数の型によって異なります。次の表に各変数型の使用メモリ量を示します。

変数型	通常変数	配列変数
整数型	5バイト	5+2×(要素数)+2×(次元数)+1バイト
単精度実数型	7バイト	5+4×(要素数)+2×(次元数)+1バイト
倍精度実数型	11バイト	5+8×(要素数)+2×(次元数)+1バイト
文字型	6+(文字数)バイト	5+3×(要素数)+2×(次元数)+1+(各要素に含まれる文字列の総文字数)バイト

3-6　式と演算

式は演算(計算)の手順を示すもので、1個以上の定数や変数を0個以上の演算子で結合したものです。式の演算結果は、1個の数値または文字列になります。また、演算子を含まない数値や文字列だけのものも式と呼びます(例：10, 3+5＊4-2, "ABC", A%＊B#+C!, "HELLO"+"WORLD")。

BASICの演算には、数値型を扱う算術演算、論理演算と、文字型を扱う文字列演算、数値型と文字型の両方を扱う関係演算、関数があります。

● 算術演算

算術演算は、数値型の変数、定数を算術演算子で結合することをいいます。文字型を扱うことはできません。次の表に算術演算子の一覧を示します。

算術演算子	演算内容	例	実行結果
^	指数演算を行う2項演算子 (左辺)の(右辺)乗を得る	2^10	1024
-	符号反転を行う単項演算子 正符号と負符号を反転する	-10	-10
＊	乗算を行う2項演算子 (左辺)と(右辺)の積を得る	3＊27	81
/	除算を行う2項演算子 (左辺)を(右辺)で割った商を得る	80/25	3.2
¥	除算を行う2項演算子 (左辺)を(右辺)で割った商を整数で得る (演算後に小数点以下切り捨て)	80¥25	3
MOD	除算を行う2項演算子 (左辺)を(右辺)で割った余りを整数で得る (演算前に小数点以下切り捨て)	80 MOD 25	5
+	加算を行う2項演算子12+34 (左辺)と(右辺)の和を得る	46	
-	減算を行う2項演算子 (左辺)から(右辺)を引いた値を得る	34-12	22
()	優先順位を指定する演算子 括弧で囲まれた演算を他の演算より 先に実行する	(3+4)/2	3.5

算術演算の結果、エラーが発生する場合があります。

- **0による除算** ―― 数値を0で割ると、エラー(Division by zero)が発生する。0に対して負のベキ乗を行った場合も同様。
- **桁あふれ** ―― 演算の結果が、演算の型で扱える範囲を超えていると、エラー(Overflow)が発生する。

● 関係演算

関係演算は、2つの数値または2つの文字列を比較することをいいます。比較の結果は、真(-1)か、偽(0)で与えられます。次の表に関係演算子の一覧を示します。

関係演算子	演算内容
=	(左辺)と(右辺)が等しければ真
<>, ><	(左辺)と(右辺)が異なっていれば真
<	(左辺)が(右辺)より小さければ真
>	(左辺)が(右辺)より大きければ真
<=, =<	(左辺)が(右辺)より小さいか等しければ真
>=, =>	(左辺)が(右辺)より大きいか等しければ真

数値型の場合、大小の基準は代数と同じですが、文字型の場合は次の規則に従います。

・文字列の先頭から文字ごとの比較を行い、すべての文字が一致すれば、2つの文字列は等しい。
・2つの文字列が一致しない場合、不一致が発生した最初の文字のキャラクタコードを比較し、その値が大きい方の文字列が大きい。
・文字ごとの比較で不一致は発生しなかったが、一方の文字列が他方よりも長い場合、長い文字列の方が大きい。

●論理演算

論理演算はブール代数に基づく演算で、複数の条件を元に判断を行う場合や、整数のビット操作を行う場合に使用します。論理演算の結果は、ビットごとに0か1で与えられます。文字型を扱うことはできません。次の表に論理演算子の一覧を示します。

	左辺	0	1	0	1	
論理演算子	演算内容　　右辺	0	0	1	1	備考
NOT	否定の単項演算子	1	1	0	0	左辺は存在しない。右辺の真偽を反転させる
AND	論理積の2項演算子	0	0	0	1	両方が真(1)のときにのみ真になる
OR	論理和の2項演算子	0	1	1	1	一方でも真(1)ならば真になる
XOR	排他論理和の2項演算子	0	1	1	0	両辺が食い違うときにのみ真(1)になる
EQV	同値の2項演算子	1	0	0	1	両辺が一致するときにのみ真(1)になる
IMP	包含の2項演算子	1	0	1	1	左辺が右辺に含まれるなら真(1)になる

論理演算では、演算前に両辺を整数型に変換し、整数のビットごとに演算を行います。もし、整数型に変換できない(-32768～32767に収まらない)値であれば、エラー(Overflow)が発生します。

なお、ここで説明した論理演算子と、COPY *2などのパラメータに使用する論理演算子は別のものです。

●文字列演算

文字列演算は、文字型の変数、定数を文字列演算子で結合することをいいます。数値型を扱うことはできません。文字列演算に定義されている演算子は、連結演算子と関係演算子です。

連結演算子は2つの文字列を連結し、単一の文字列を得ます。演算子の記号はプラス(+)です(例："ABC"+"DEF" → "ABCDEF")。

関係演算子については、前述の「関係演算」の項を参照してください。

●関数

関数は、与えられた引数に対して特定の演算を行い、その結果を返すものです。BASICには、組み込み関数としてABSやSQRなどの数値型関数や、CHR$やMID$などの文字型関数が用意されています。

また、ユーザーが関数を定義することもできます。詳細はDEF FN文の説明を参照してください。

●演算の優先順位

演算は、次の順位によって行われます。一覧の上にあるものほど順位が高く、演算が優先して行われます。

括弧で囲まれた演算
関数
指数演算(2項演算子 ^)
符号反転(単項演算子 -)
乗算、除算(2項演算子 *, /)
整数除算(2項演算子 ¥)
除算の余り(2項演算子 MOD)
加算、減算(2項演算子 +, -)
関係演算(2項演算子 =, <>, ><, <, >, <=, =<, >=, =>)
否定(単項演算子 NOT)
論理積(2項演算子 AND)
論理和(2項演算子 OR)
排他論理和(2項演算子 XOR)
包含(2項演算子 IMP)
同値(2項演算子 EQV)

4. エラーメッセージ

プログラムの実行を続けられないようなエラーが発生すると、BASICはプログラムの実行を中断し、エラーメッセージを表示してコマンドレベルに戻ります。このとき、ダイレクトモードでのエラーではエラーメッセージのみが表示されますが、プログラムモードの場合は実行を中断した行の行番号を"<エラーメッセージ> in <行番号>"の形式で表示します(例：Syntax error in 10)。

メッセージの意味については、エラーコード表を参照してください。

5. 画面の座標系

画面上の位置を指定するための座標系は、各スクリーンモードによってまちまちです(解像度が異なるため)。しかし、画面左上を原点(0, 0)とし、画面右、画面下へ向かうに従ってX座標、Y座標座標が大きくなっていくのはすべてのスクリーンモードで共通です。

各スクリーンモードの詳細については、SCREEN文を参照してください。

6. スプライト

MSXには、通常の画面の他にスプライト面と呼ばれる画面が存在しています。スプライト面は通常画面の上に重なって存在しており、通常画面と干渉することなく8×8ドットか16×16ドットのパターンを表示することができます。

スプライトの詳細については、PUT SPRITEやSPRITE$を参照してください。

7. ファイル名

MSXで扱うファイルの名前(ファイル名)は、8文字以内の名前と3文字以内の拡張子からなる文字列で、名前と拡張子の間はピリオド(.)で区切ります。使用できる文字は、英数字、片仮名、平仮名、記号(- _ # $ % & ( ) @ ' ` ^ { } ~)です。

カセットテープの場合、ファイル名は拡張子なしの6文字以内で指定します。

コマンドリファレンス

コマンド書式の凡例

命令名：その命令のBASICでの表記です。一つの命令が複数の意味を持つ場合は、番号を付けて必要な数を記述しています。

読み：命令名の標準的な読み方です。

種別：命令の種類です。次のように分類しています。
- ステートメント ──→ 値を持たない
- 関数 ──→ 値を持つが代入は不可
- システム ──→ 変数値を持ち代入が可能

環境：命令を実行するのに最低限必要な環境を示します。
環境の指定がない場合は、MSX1以降
- MSX2 ──→ MSX2以降
- MSX2+ ──→ MSX2+以降
- Disk ──→ Disk BASIC v1以降
- Disk2 ──→ Disk BASIC v2以降
- MUSIC ──→ MSX-MUSIC

COPY *2 (コピー)ステートメント　MSX2　Disk

機能……その命令の概要です。

書式……その命令の記述方法を表します。<>で囲まれた部分がパラメータです。
[]で囲まれた部分は省略可能です。…はそのパラメータを繰り返し記述できることを意味します。

文例……書式に従った命令の使用例です。

解説……命令の詳細やパラメータの意味が記述されています。

参照……その命令と関連する命令の一覧が記述されています。

ABS(アブソリュート)関数

機能 絶対値を得る

書式 ABS(<数式>)

文例 PRINT ABS(−3.14)
実行結果　3.14

解説 <数式>の絶対値を得ます。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数型になります。

ASC(アスキー)関数

機能 文字のキャラクタコードを得る

書式 ASC(<文字式>)

文例 PRINT ASC("A")
実行結果　65 (65=&H41)

解説 <文字式>の最初の文字のキャラクタコードを得ます。得られる値は整数型です。<文字式>がヌルストリング("")の場合、エラー(Illegal function call)となります。

文字とキャラクタコードの対応については、キャラクタコード表を参照してください。なお、グラフィックキャラクタのキャラクタコードはすべて1となります。

参照 CHR$, キャラクタコード表

● サンプルプログラム

```
10 PRINT "Push any key!"
20 I$=INKEY$:IF I$="" GOTO 20
30 I=ASC(I$):PRINT "'"I$"' ="I"(&H"HEX$(I)")"
40 GOTO 10
```

ATN(アークタンジェント)関数

機能 逆正接(アークタンジェント)を得る

書式 ATN(<数式>)

文例 PRINT ATN(1)＊4
実行結果　3.1415926535898

解説 <数式>の逆正接(アークタンジェント)を、ラジアン単位で得ます。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は$-\pi/2$～$\pi/2$の倍精度実数になります。

参照 TAN, SIN, COS

AUTO(オート)ステートメント

機能 行番号を自動的に発生させる

書式 AUTO [<行番号>], [<増分>]

文例 AUTO 100,10

解説 AUTOを実行すると、<行番号>で指定した行番号を表示してプログラムの入力待ちになり、以後、[RETURN]を入力するごとに<増分>ずつ増加した行番号を表示します。<行番号>のみを省略すると0が、<増分>のみを省略すると10が、両方を省略するとそれぞれ10が指定されたとみなされます。

AUTOによる行番号の自動発生を止めるには、[CTRL]+[STOP]か[CTRL]+[C]を入力します。

既存の行番号と同じ行番号を発生した場合、行番号の後ろにアスタリスク(＊)が表示されます。このとき、何も入力せずに[RETURN]を押すと元の行は保存されます。

参照 RENUM, LIST

BEEP(ビープ)ステートメント

機能 スピーカを鳴らす

書式 BEEP

文例 BEEP

解説 MSX1では、スピーカを約0.04秒鳴らします。MSX2以降では、音色と音量をSET BEEP文で指定できます。また、PRINT CHR$(7)を実行しても、BEEP文と同じ結果が得られます。

参照 SET BEEP

BIN$(バイナリ ダラー)関数

機能 数値を2進表記の文字列に変換し、その結果を得る

書式 BIN$(<数式>)

文例 PRINT BIN$(100)
実行結果　1100100

解説 <数式>の値を2進表記の文字列に変換し、その結果を得ます。得られる値は文字型です。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、2進表記への変換は整数型への変換後に行われます。そのため、<数式>に指定できる値の範囲は-32768～32767(または0～65535)となります。

参照 VAL, OCT$, HEX$, STR$

BLOAD [*1] (ビーロード)ステートメント

機能 機械語プログラムをメモリに読み込む

書式 BLOAD <ファイルスペック> [, R] [, <オフセット>]

文例 BLOAD "MACHINE.BIN",R

解説 <ファイルスペック>で指定するディスク上のファイルをメモリに読み込みます。読み込めるのは、BSAVE [*1]かBSAVE [*2]で保存したファイルだけです。読み込みを行う番地は、BSAVE [*1]で指定した開始番地に<オフセット>を加えた値になります。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブのファイルを読み出します(Disk BASICが存在する場合)。

Rオプションを付けると、メモリに読み込んだデータを機械語プログラムとみなして実行を開始します。実行を開始する番地は、BSAVE [*1]で指定した実行開始番地に<オフセット>を加えた値になります。

<オフセット>には、0～65535の整数を指定します。<オフセット>を省略すると、0が指定されたものとみなされます。

参照 BSAVE [*1], USR

BLOAD [*2] (ビーロード)ステートメント　Disk

機能 画面データをVRAMに読み込む

書式 BLOAD <ファイルスペック> ,S [, <オフセット>]

文例 BLOAD "SAMPLE.SC7",S

解説 <ファイルスペック>で指定するディスク上のファイルをVRAMに読み込みます。読み込めるのは、BSAVE [*2]かBSAVE [*1]で保存したファイルだけです。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブから読み込みます。

<オフセット>は、アクティブページ先頭からのオフセットで、0～65535の整数を指定します。<オフセット>を省略すると、0が指定されたものとみなされます。

参照 BSAVE [*2], COLOR=RESTORE, SET PAGE

BSAVE [*1] (ビーセーブ)ステートメント

機能 メモリ上のデータをファイルに保存する

書式 BSAVE <ファイルスペック> ,<開始番地> ,<終了番地> [, <実行開始番地>]

文例 BSAVE "MACHINE.BIN",&HA100,&HB000

解説 <開始番地>と<終了番地>で指定するメモリ上の領域を、<ファイルスペック>で指定するファイルに保存します。<実行開始番地>には、保存したデータを機械語プログラムとして扱う場合の実行開始番地を指定します。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブのファイルを読み出します(Disk BASICが存在する場合)。

<開始番地>、<終了番地>、<実行開始番地>には、&H8000～&HFFFFの整数を指定します。<実行開始番地>を省略すると、<開始番地>と同じ値が指定されたものとみなされます。

参照 BLOAD [*1]

BSAVE [*2] (ビーセーブ)ステートメント　Disk

機能 VRAM上の画面データを保存する

書式 BSAVE <ファイルスペック> ,<開始番地> ,<終了番地> ,S

文例 BSAVE "SAMPLE.SC7",&H0000,&HFA9F,S

解説 <開始番地>と<終了番地>で指定するVRAMの領域を、<ファイルスペック>で指定するディスク上のファイルに保存します。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブに保存します。

<開始番地>と<終了番地>は、共にアクティブページ先頭からのオフセットで、0～65535の整数を指定します。

参照 BLOAD [*2], SET PAGE

CALL BGM(コールビージーエム)ステートメント　MUSIC

機能 MSX-MUSICの演奏をバックグラウンドで行うように設定する

書式 CALL BGM(<モード>)

文例 CALL BGM(0)

解説 MSX-MUSICによる演奏をバックグラウンドで行うかどうかを設定します。<モード>に0を指定すると、演奏中に他の命令を実行しない設定になります。1を指定すると、演奏中に他の命令を実行する設定になります(バックグラウンド演奏)。CALL MUSIC文によって初期化された直後は、バックグラウンド演奏に設定されています。

参照 CALL MUSIC, PLAY#, CALL STOPM

CALL MEMINI(コール メモリ イニシャライズ)ステートメント　MSX2

機能 メモリディスクを初期化する

書式 CALL MEMINI[(<メモリディスクの上限>)]

文例 CALL MEMINI

解説 BASICが使用しないメモリをメモリディスクとして使用するための初期化処理を行います。

メモリディスクのデバイス名は"MEM:"です。

<メモリディスクの上限>に指定する値によって、メモリディスクの容量が変わります。<メモリディスクの上限>に指定する値の範囲は&H0000～&H7FFFですが、&H0000～&H03FEを指定するとメモリディスクを削除します。<メモリディスクの上限>を省略すると、&H7FFFが指定されたものとして動作します。

参照 CALL MFILES, CALL MKILL, CALL MNAME

CALL MFILES(コールエムファイルズ)ステートメント MSX2

機能 **メモリディスク上に存在するファイルの一覧を表示する**

書式 CALL MFILES

文例 CALL MFILES

解説 メモリディスク上に存在するファイルの一覧を表示します。

参照 CALL MEMINI, CALL MKILL, CALL MNAME

CALL MKILL(コールエムキル)ステートメント MSX2

機能 **メモリディスク上のファイルを消去する**

書式 CALL MKILL(<ファイル名>)

文例 CALL MKILL("TEST.BAS")

解説 <ファイル名>で指定するメモリディスク上のファイルを消去します。<ファイル名>にワイルドカードを使用することはできません。

参照 CALL MEMINI, CALL MFILES, CALL MNAME

CALL MNAME(コールエムネーム)ステートメント MSX2

機能 **メモリディスク上のファイル名を変更する**

書式 CALL MNAME(<旧ファイル名> AS <新ファイル名>)

文例 CALL MNAME("TEMP.DAT" AS "USER.DAT")

解説 <旧ファイル名>で指定するメモリディスク上のファイルのファイル名を、<新ファイル名>に変更します。<旧ファイル名>、<新ファイル名>にワイルドカードを使用することはできません。

参照 CALL MEMINI, CALL MFILES, CALL MKILL

CALL MUSIC(コールミュージック)ステートメント MUSIC

機能 **MSX-MUSICを初期化する**

書式 CALL MUSIC[([<モード>] [, [0]] [, <PLAY#文第1文字式のチャンネル数>] [, <PLAY#文第2文字式のチャンネル数> …])]

文例 CALL MUSIC(0,,1,1,1,1,1,1,1,1,1)
実行結果 MSX-MUSICをリズム音なしのモードで初期化し、すべてのチャンネルをPLAY#文から制御できるようにする

解説 MSX-MUSICを初期化し、各チャンネルの割り振りを行います。CALL MUSICによって初期化を行うまで、MSX-MUSICの諸命令を使用することはできません。

<モード>には、リズム音を使用するかどうかを指定します。0を指定するとリズム音を使用せず、9チャンネルすべてで楽音を使用できます。1を指定するとリズム音を使用するモードとなり、楽音に使用できるチャンネル数は6になります。

第2パラメータはMSX-AUDIOとの互換性を保つために存在し、常に0を指定します。

第3パラメータ以降には、PLAY#文に指定する文字式へ割り振るチャンネル数を、チャンネル番号の小さい方から割り当てます。チャンネル数に1以外の値を指定すると、複数のチャンネルで一つの文字式を演奏します。チャンネル数の合計は、6(リズム音を使用するモード)または9(リズム音を使用しないモード)以下でなければなりません。また、チャンネル数に0を指定することはできません。

すべてのパラメータを省略した場合、CALL MUSIC(1,0,1,1,1)と同じ設定になります(リズム音を使用するモードでPLAY#文の第1～3文字式にチャンネル1～3を割り当てる)。

CALL MUSICを実行すると、すべての変数が初期化され、BASICの使用可能なメモリが807バイト減少します。また、システムの割り込み処理にMSX-MUSICの処理が追加されるため、パフォーマンスが低下します。

参照 CALL BGM, CALL PITCH, PLAY#, CALL PLAY, CALL STOPM, CALL TEMPER, CALL TRANSPOSE, CALL VOICE, CALL VOICE COPY

CALL PITCH(コールピッチ)ステートメント MUSIC

機能 **MSX-MUSICの音高(ピッチ)を調節する**

書式 CALL PITCH(<ピッチ1> [, <ピッチ2>])

文例 CALL PITCH(440)

解説 MSX-MUSIC全体の音高(ピッチ)を調節します。<ピッチ1>に"O4A"の周波数を指定することで音高を調節します。<ピッチ1>の範囲は410～459で、単位はHzです。初期値は440です。音高はトランスポーズと独立に設定できます。

<ピッチ2>は、MSX-AUDIOとの互換性のために存在し、MSX-MUSICでは指定しても無視されます。

参照 CALL MUSIC, PLAY#, CALL TEMPER, CALL TRANSPOSE

CALL PLAY(コールプレイ)ステートメント MUSIC

機能 **MSX-MUSICの演奏状態を調べる**

書式 CALL PLAY(<文字式番号>, <変数名>)

文例 CALL PLAY(1,A)

解説 <文字式番号>で指定するPLAY#文の文字式が演奏中かどうかを調べ、その結果を<変数名>で指定する変数に格納します。演奏中であれば真(-1)を、そうでなければ偽(0)を得ます。

<文字式番号>は、PLAY#文の先頭の文字式を1とし、右へ行くに従って1ずつ増加する整数値です。<文字式番号>に0を指定すると、演奏中の文字式が一つでもあれば真(-1)を得ます。

参照 CALL MUSIC, PLAY#

CALL STOPM(コールストップミュージック)ステートメント MUSIC

機能 **MSX-MUSICのバックグラウンド演奏を停止する**

書式 CALL STOPM

文例 CALL STOPM

解説 MSX-MUSICによるバックグラウンド演奏を停止します。

参照 CALL MUSIC, PLAY#, CALL BGM

CALL SYSTEM(コールシステム)ステートメント Disk

機能 **MSX-DOSを起動する**

書式 CALL SYSTEM

文例 CALL SYSTEM

解説 BASICを終了し、MSX-DOSを起動します。MSX-DOSを起

動すると、すべてのファイルは閉じられ、メモリ上のプログラム、データとメモリディスク上のファイルは失われます。

この命令は、MSX-DOSからBASICを起動した場合にのみ、実行可能です。

CALL TEMPER(コールテンペラメント)ステートメント　MUSIC

機能 MSX-MUSICで音律を設定する

書式 CALL TEMPER(<音律番号>)

文例 CALL TEMPER(0)

解説 MSX-MUSICの音律を、<音律番号>で指定する値に設定します。音律は、1オクターブを12分割する際の比率で、楽音の音高に影響を与えます。指定できる音律は下表の通りです。初期値は、9番の完全平均律です。

音律番号	音律名
0	ピタゴラス
1	ミーントーン
2	ヴェルクマイスター
3	ヴェルクマイスター(修正)
4	ヴェルクマイスター(別)
5	キルンベルガー
6	キルンベルガー(修正)
7	ヴァロッティ・ヤング
8	ラモー
9	完全平均律
10	純正律cメジャー(aマイナー)
11	純正律cisメジャー(b)
12	純正律dメジャー(h)
13	純正律esメジャー(c)
14	純正律eメジャー(cis)
15	純正律fメジャー(d)
16	純正律fisメジャー(es)
17	純正律gメジャー(e)
18	純正律gisメジャー(f)
19	純正律aメジャー(fis)
20	純正律bメジャー(g)
21	純正律hメジャー(gis)

参照 CALL MUSIC, PLAY#, CALL PITCH, CALL TRANSPOSE

CALL TRANSPOSE(コールトランスポーズ)ステートメント　MUSIC

機能 MSX-MUSICで移調を行う

書式 CALL TRANSPOSE(<トランスポーズ値1> [, <トランスポーズ値2>])

文例 CALL TRANSPOSE(0)

解説 <トランスポーズ値1>にセント(半音の1/100)単位の値を与えて、移調を行います。<トランスポーズ値1>の範囲は-12799~+12799で、初期値は0です。トランスポーズは音高(ピッチ)と独立に設定できます。

<トランスポーズ値2>は、MSX-AUDIOとの互換性のために存在し、MSX-MUSICでは指定しても無視されます。

参照 CALL MUSIC, PLAY#, CALL TEMPER, CALL PITCH

CALL VOICE(コールボイス)ステートメント　MUSIC

機能 MSX-MUSICのチャンネルに音色を設定する

書式 CALL VOICE([<チャンネル1の音色>] [, [<チャンネル1の音色>] …])

文例 CALL VOICE(,,@1)

解説 MSX-MUSICの各チャンネルに音色を割り当てます。割り当ては、@ <数式>の形式で音色番号を指定するか、音色情報を格納した配列変数の変数名を指定して行います。指定を省略したチャンネルの音色は変更されません。音色番号を指定する場合は、次の表から演奏に使用する音色を選択してください。

音色番号	音色名	内蔵音色	音色番号	音色名	内蔵音色
0	Piano 1	○	32	Piano 3	
1	Piano 2		33	Electric Piano 2	○
2	Violin	○	34	Santool 2	
3	Flute 1	○	35	Brass	
4	Clarinet	○	36	Flute 2	
5	Oboe	○	37	Clavicode 2	
6	Trumpet	○	38	Clavicode 3	
7	Pipe Organ 1		39	Koto 2	
8	Xylophone		40	Pipe Organ 2	
9	Organ	○	41	PohdsPLA	
10	Guitar	○	42	RohdsPRA	
11	Santool 1		43	Orch L	
12	Electric Piano 1	○	44	Orch R	
13	Clavicode 1		45	Synthesizer Violin	
14	Harpsicode 1	○	46	Synthesizer Organ	
15	Harpsicode 2		47	Synthesizer Brass	
16	Vibraphone 1	○	48	Tube	○
17	Koto 1		49	Shamisen	
18	Taiko		50	Magical	
19	Engine 1		51	Huwawa	
20	UFO		52	Wander Flat	
21	Synthesizer Bell		53	Hardrock	
22	Chime		54	Machine	
23	Synthesizer Bass	○	55	Machine V	
24	Synthesizer	○	56	Comic	
25	Synthesizer Percussion		57	SE-Comic	
26	Synthesizer Rhythm		58	SE-Laser	
27	Harm Drum		59	SE-Noise	
28	Cawbell		60	SE-Star 1	
29	Close Hi-hat		61	SE-Star 2	
30	Snare Drum		62	Engine 2	
31	Bass Drum		63	Silence	

各音色のうち、内蔵音色の欄に印のないものはオリジナル音色と呼ばれ、同時に1音しか指定できません。複数のオリジナル音色を指定すると、CALL VOICE文の最後のパラメータか、最後に実行したCALL VOICE文の設定が使われます。

音色情報は32バイトのデータです。フォーマットを次の表に示します。

オフセット	分類	内容
0~7	ヘッダ	音色名
8~9		ボイス移調
10		bit1~3: フィードバック
11~15		予約

次ページへ続く

オフセット	分 類	内 容
16	オペレータ0	bit0～3: MULT
		bit4: KSR
		bit5: EG
		bit6: PM
		bit7: AM
17		bit0～5: トータルレベル
		bit6～7: レベルキースケール
18		bit0～3: ディケイレイト
		bit4～7: アタックレイト
19		bit0～3: リリースレイト
		bit4～7: サスティンレベル
20～23		予約
24	オペレータ1	bit0～3: MULT
		bit4: KSR
		bit5: EG
		bit6: PM
		bit7: AM
25		bit6～7: レベルキースケール
26		bit0～3: ディケイレイト
		bit4～7: アタックレイト
27		bit0～3: リリースレイト
		bit4～7: サスティンレベル
28～31		予約

音色情報で指定する音色もオリジナル音色の一種であり、同時に使用できるのは1音だけです。

参照 CALL MUSIC, CALL VOICE COPY, PLAY#

CALL VOICE COPY(コールボイスコピー)ステートメント MUSIC

機能 音色情報を転送する

書式 CALL VOICE COPY(<転送元>, <転送先>)

文例 CALL VOICE COPY(A,@63)

解説 <転送元>で指定する音色情報を、<転送先>へ複製します。<転送元>には、@<数式>形式で指定する音色番号か、音色情報を格納した配列変数の変数名を指定します。<転送先>には、@63か音色情報を格納する配列変数の変数名を指定します。<転送元>に指定できる音色番号は、内蔵音色(CALL VOICE参照)でないものに限られます。

音色情報は32バイトのデータです。音色情報の詳細については、CALL VOICEを参照してください。

参照 CALL MUSIC, CALL VOICE

CDBL(シーダブル)関数

機能 数値型の値を倍精度実数に変換し、その結果を得る

書式 CDBL(<数式>)

文例 A#=CDBL(3.14)

解説 <数式>の値を倍精度実数に変換し、その結果を得ます。ただし、型変換が行われるだけで有効桁数は変化しないため、精度は変換前と同じになります。

CDBLで得られる値は、数値型の値を倍精度実数型変数に代入した場合と同じです。

参照 CSNG, CINT

CHR$(キャラクタダラー)関数

機能 指定したキャラクタコードに対応する文字を得る

書式 CHR$(<数式>)

文例 PRINT CHR$(&H41)

実行結果 A

解説 <数式>の値をキャラクタコードとして、対応する文字を得ます。得られる値は文字型です。<数式>に指定できる値の範囲は0～255です。これ以外の値を指定すると、エラー(Illegal function call)となります。

キャラクタコードと文字の対応については、キャラクタコード表を参照してください。なお、グラフィックキャラクタについてはキャラクタコードの前にCHR$(1)を付加することで取得できます。

参照 ASC, キャラクタコード表

● サンプルプログラム

```
10 CLS:PRINT "Normal character table"
20 FOR I=0 TO 15:FOR J=2 TO 15
30 PRINT CHR$(J*16+I)" ";
40 NEXT:PRINT:NEXT
50 I$=INPUT$(1)
60 CLS:PRINT "Graphic character table"
70 FOR I=0 TO 15:FOR J=4 TO 5
80 PRINT CHR$(1)CHR$(J*16+I)" ";
90 NEXT:PRINT:NEXT
```

CINT(シーインテジャー)関数

機能 数値型の値を整数に変換し、その結果を得る

書式 CINT(<数式>)

文例 PRINT CINT(3.14)

実行結果 3

解説 <数式>の値を整数に変換し、その結果を得ます。型変換時に小数点以下は切り捨てられます。変換結果が-32768～32767の範囲外になる場合は、エラー(Overflow)となります。

CINTで得られる値は、数値型の値を整数型変数に代入した場合と同じです。

参照 CSNG, CDBL

CIRCLE(サークル)ステートメント

機能 円または楕円を描く

書式 CIRCLE [STEP] (x, y), <半径> [, <カラーコード>] [, <開始角度>] [, <終了角度>] [, <比率>]

文例 CIRCLE (128,106),80,15,3.14/2,3.14＊2,2

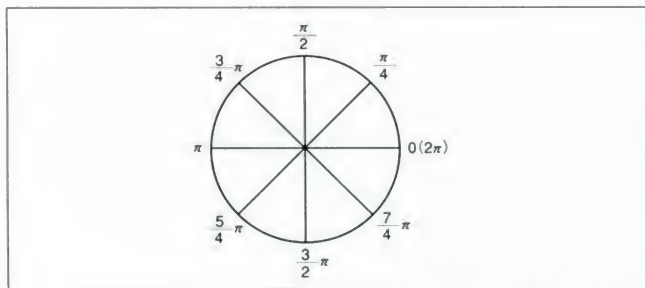
解説 座標(x, y)を中心とし、<半径>で指定する大きさを持つ円を描きます。グラフィックモード(SCREEN 2～12)でのみ使用できます。(x, y)に指定可能な値を次の表に示します。

グラフィックモード	x	y
SCREEN 2～4	0～255	0～191
SCREEN 5	0～255	0～211
SCREEN 6～7	0～511	0～211
SCREEN 8～12	0～255	0～211

(x, y)を最終参照点(LP)からの相対座標で指定する場合は、STEPを(x, y)の前に付加します。LPはCIRCLE文の実行後、(x, y)に設定されます。

<カラーコード>を指定すると、指定した色で円が描かれます。<カラーコード>を省略した場合は、COLOR文で指定した前景色が用いられます。

<開始角度>、<終了角度>を指定すると、指定した角度(単位はラジアンで-2π～2πを指定可能)の範囲のみ、円弧が描かれます。負の角度を指定した場合はその絶対値が用いられ、扇形が描かれます。<開始角度>、<終了角度>を省略した場合、それぞれ0、2πが用いられます。

<比率>には、(垂直方向の半径) / (水平方向の半径)を指定します。省略時値は1.0です。<比率>が1未満の場合は垂直方向の半径が短くなり、1を超える場合は水平方向の半径が短くなります。

参照 COLOR, SCREEN, LINE, PSET, PRESET, PAINT

● サンプルプログラム

```
10 COLOR 15,1,1:SCREEN 2
20 CIRCLE (100,50),80,15,0,3.14,.5
30 FOR I=0 TO 8
40 CIRCLE STEP (10,10),30,12,,,.75
50 NEXT
60 I$=INPUT$(1)
```

CLEAR(クリア)ステートメント

機能 変数を初期化し、メモリ領域の大きさを設定する

書式 CLEAR [<文字領域の大きさ> [, <BASICの管理領域の上限>]]

文例 CLEAR 500,&HD000

解説 すべての変数を初期化し、BASICの文字領域の大きさと管理領域の上限を設定します。

CLEAR文を実行すると、すべての数値型変数は0に、文字型変数はヌルストリング("")に初期化され、DEFINT、DEFSNG、DEFDBL、DEFSTR、DEF FNやON ERROR GOTO、ON INTERVAL GOSUB、ON KEY GOSUB、ON SPRITE GOSUB、ON STOP GOSUB、ON STRIG GOSUBによる定義は無効になります。また、すべてのファイルが閉じられます。

<文字領域の大きさ>には、BASICが文字列を格納する領域の大きさを整数で指定します。起動直後は、200に設定されています。

<BASICの管理領域の上限>には、BASICが使用するメモリ領域の上限を整数(&H831F～&HF380)で指定します。BASICは指定した値の直前までを使用するため、ユーザーは指定値からシステム領域の直前までを自由に使用できます。起動直後は、システム領域の先頭に設定されています。なお、システム領域はDisk BASICが存在しない場合は&HF380から始まりますが、存在する場合はHIMEM(&HFC4A、2byte)で指定される番地からとなります。

<文字領域の大きさ>、<BASICの管理領域の上限>を省略すると、以前の設定値をそのまま使用します。

参照 FRE

CLOSE(クローズ)ステートメント

機能 ファイルを閉じる

書式 CLOSE [[#] <ファイル番号> [, [#] <ファイル番号> …]]

文例 CLOSE #1

解説 <ファイル番号>に対応するファイルを閉じます。<ファイル番号>を複数指定することにより、同時に複数のファイルを閉じることができます。また、<ファイル番号>を省略すると、そのとき開いているすべてのファイルを閉じます。閉じた<ファイル番号>は、以後のOPEN文で自由に指定できます。

CLOSE文は出力バッファに残っているデータの掃出しを行います。出力用に開いたファイルは、必ずCLOSE文を実行して出力処理を完了してください。また、END文、NEW文を実行すると、すべてのファイルが閉じられますが、STOP文はファイルを閉じないので注意してください。

参照 OPEN, PRINT#, INPUT#, NEW, END, MAX FILES

CLS(クリア スクリーン)ステートメント

機能 画面に表示されている文字や絵を消去する

書式 CLS

文例 CLS

解説 画面に表示されている文字や絵をすべて消去します。ただし、スプライトパターン表示と、KEY ON *1状態の時に表示される画面下のファンクションキー表示は例外です。

テキストモード(SCREEN 0～1)でCLS文を実行すると、カーソル位置は(0,0)へ移動します。グラフィックモード(SCREEN 2～12)で実行した場合、最終参照点(LP)は保存されます。

参照 SCREEN, KEY ON *1, KEY OFF *1, PUT SPRITE, COLOR

COLOR(カラー)ステートメント

機能 画面の前景色、背景色、周辺色を設定する

書式 COLOR [<前景色>] [, <背景色>] [, <周辺色>]

文例 COLOR 15,4,7

解説 <前景色>、<背景色>、<周辺色>をカラーコードで指定します。カラーコードに指定できる値の範囲は、SCREEN 6が0～3、SCREEN 8/11/12が0～255、それ以外のスクリーンモードでは0～15の整数です。指定を省略した場合、省略されたパラメータはCOLOR文を実行する前の状態を保ちますが、3つの指定すべてを省略することはできません(すべて省略した場合、MSX2以降ではエラーになりませんが、COLOR=NEW文の省略形を実行したとみなされます)。

起動直後には、それぞれ15(白)、4(暗い青)、7(水色)が設定されています。

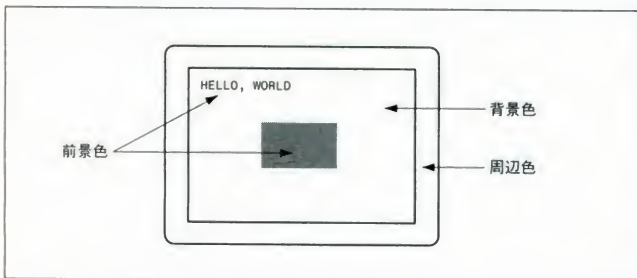
MSX2以降では、512色の中からカラーコードに色を割り当てることができます(COLOR=を参照してください)。MSX1および初期状態のMSX2／2+／turboRでは、各カラーコードに次の色が割り当てられています(括弧内はそれぞれ赤、緑、青のレベルを表しています)。

0: 透明(0,0,0)　1: 黒(0,0,0)　2: 緑(1,6,1)　3: 明るい緑(3,7,3)
4: 暗い青(1,1,7)　5: 明るい青(2,3,7)　6: 暗い赤(5,1,1)　7: 水色(2,6,7)
8: 赤(7,1,1)　9: 明るい赤(7,3,3)　10: 黄(6,6,1)　11: 明るい黄(6,6,4)
12: 暗い緑(1,4,1)　13: 紫(6,2,5)　14: 灰(5,5,5)　15: 白(7,7,7)

<前景色>は、テキストモード(SCREEN 0～1)で文字を表示するときや、グラフィックモード(SCREEN 2以上)で点や線を描くときに使われる色です。PSET, LINE, CIRCLE, PAINTなどは、特に指定しない限り、<前景色>を用いて描画を行います。

<背景色>は、画面の背景になる色です。テキストモードであればCOLOR文の実行後すぐに、グラフィックモードであればCOLOR文後のSCREEN文かCLS文の実行後に指定したカラーコードに変わります。

<周辺色>は、表示領域外の画面周辺部の色です。<前景色>、<背景色>に透明色が指定された場合、<周辺色>が表示されます。ただし、SCREEN 0に限り<周辺色>は無視され、<背景色>と同じになります。

<前景色>、<背景色>、<周辺色>に4以上の値を指定した状態でSCREEN 6に移行すると、各カラーコードを4で割った余りが指定されたものとして扱われます。また、SCREEN 6で指定したカラーコード0～3は他のスクリーンモードに移行すると、0、5、10、15に変わります。

参照 COLOR=, SCREEN, CLS, LINE, CIRCLE, PSET, PAINT, PRESET

COLOR=(カラー イコール)ステートメント　MSX2

機能 カラーコードの色を変更する

書式 COLOR=(<カラーコード>, <Rコード>, <Gコード>, <Bコード>)

文例 COLOR=(3,7,7,7)

解説 各<カラーコード>に512色中の1色を割り当てます。<カラーコード>には0～15の整数を指定します(SCREEN 6のみ0～3)。ただし、カラーコード0は透明色に固定されています。

<Rコード>、<Gコード>、<Bコード>にはそれぞれ赤、緑、青の成分を0～7の整数で指定します。

COLOR=文で割り当てた色はCOLOR=NEW文の実行、スクリーンモードの変更によって無効となり、初期状態に戻ります。

COLOR=文によるカラーコードの指定は、SCREEN 8/11/12では意味を持ちません。SCREEN 8では、下記のようにカラーコードが直接、色を表し、SCREEN 11/12ではVRAMの値そのものです。

7	6	5	4	3	2	1	0
G			R			B	

G: 緑の成分(0～7)
R: 赤の成分(0～7)
B: 青の成分(0～3)
SCREEN 8のカラーコード = G × 32 + R × 4 + B

参照 COLOR, COLOR=NEW, SCREEN

● サンプルプログラム

・COLOR=を使ってフェードアウト処理を行う

```
10 SCREEN 0:COLOR 15,1,1
20 PRINT "hello, world."
30 FOR I=7 TO 0 STEP -1
40 FOR J=0 TO 4
50 T=TIME
60 IF T=TIME GOTO 60
70 NEXT
80 COLOR=(15,I,I,I)
90 NEXT
100 I$=INPUT$(1)
110 COLOR=NEW
```

・SCREEN 8で256色を同時に表示する

```
10 SCREEN 8
20 FOR R=0 TO 7:FOR G=0 TO 7:FOR B=0 TO 3
30 C = G * 32 + R * 4 + B
40 LINE (C,0) - (C,211),C
50 NEXT:NEXT:NEXT
60 I$=INPUT$(1)
```

COLOR=NEW(カラー イコール ニュー)ステートメント　MSX2

機能 カラーコードの色を初期化する

書式 COLOR[=NEW]

文例 COLOR=NEW

解説 COLOR=文で割り当てた色を初期状態に戻します。

参照 COLOR=, SCREEN

COLOR=RESTORE(カラー イコール リストア)ステートメント　MSX2

機能 VRAM上のパレットテーブルに従ってカラーコードの色を設定する

書式 COLOR=RESTORE

文例 COLOR=RESTORE

解説 VRAM上のパレットテーブルに従ってカラーコードの色を設定します(0～15すべて)。

BSAVEで保存した画像をBLOADで読み込んだ際に、COLOR=RESTOREを実行することにより、保存時と同じ色で表示を行えます。

参照 COLOR=, SCREEN, BSAVE [*2], BLOAD [*2]

COLOR SPRITE(カラー スプライト)ステートメント　MSX2

機能 スプライト面の色を指定する

書式 COLOR SPRITE(<スプライト面番号>)=<カラーコード>

文例 COLOR SPRITE(0)=15

解説 <スプライト面番号>のスプライト面に表示されるスプライト

パターンの色を、<カラーコード>に設定します。SCREEN 4以上でのみ使用できます。<カラーコード>に指定できる値については、COLOR SPRITE$を参照してください。

PUT SPRITE文でカラーコードを指定すると、COLOR SPRITEおよびCOLOR SPRITE$による指定は無効になります。

参照 COLOR SPRITE$, PUT SPRITE

● サンプルプログラム

```
10 COLOR 15,1,1:SCREEN 4
20 SPRITE$(1)=STRING$(8,CHR$(255))
30 COLOR SPRITE(10)=15
40 PUT SPRITE 10,(0,0),,1
50 I$=INPUT$(1)
60 COLOR SPRITE(10)=3
70 I$=INPUT$(1)
```

COLOR SPRITE$(カラースプライトダラー)ステートメント MSX2

機能 スプライト面の色を1行ごとに指定する

書式 COLOR SPRITE$(<スプライト面番号>)=<カラーコード列>

文例 COLOR SPRITE$(0)=STRING$(16,&H48)

解説 <スプライト面番号>のスプライト面に表示されるスプライトパターンの色を、1行ごとに設定します。<カラーコード>には下記の値を指定できます。

P: 1のとき、次のような動作になります。

- P=1が指定された行は、より小さいスプライト面番号を持つスプライト面の同一走査線上にP=0のパターンがある場合にのみ表示される
- P=0が指定された行とP=1が指定された行(複数可)のカラーコードは、各々のカラーコードの論理和になる
- 上記の重なり部分では、衝突検出が行われない
- P=0のパターンを持つスプライト面とP=1のスプライト面のスプライト面番号が連続している場合、P=0のスプライト面に対してPUT SPRITEを実行すると、P=1のスプライト面を同時に動かすことができる

I: 1のとき、指定された行の衝突検出が行われなくなります。

C:カラーコード(0～15)を指定します。SCREEN 8では下記の16色に固定されています(括弧内はそれぞれ赤、緑、青のレベルを表しています)。

0: 黒(0,0,0)	1: 暗い青(0,0,2)	2: 暗い赤(3,0,0)	3: 暗い紫(3,0,2)
4: 暗い緑(0,3,0)	5: 暗い水色(0,3,2)	6: 暗い黄色(3,3,0)	7: 灰色(3,3,2)
8: 肌色(7,4,2)	9: 青(0,0,7)	10: 赤(7,0,0)	11: 紫(7,0,7)
12: 緑(0,7,0)	13: 水色(0,7,7)	14: 黄(7,7,0)	15: 白(7,7,7)

また、SCREEN 6ではハードウェアタイリング(偶数番目のドットと奇数番目のドットに異なる色を表示する)機能により、擬似的に10色を使用できます。

7	6	5	4	3 2	1 0
−	P	I	−	C1	C2

C1: 奇数番目のドットのカラーコード(0～3)

C2: 偶数番目のドットのカラーコード(0～3)

PUT SPRITE文でカラーコードを指定すると、COLOR SPRITEおよびCOLOR SPRITE$による指定は無効になります。

参照 COLOR SPRITE, PUT SPRITE, SPRITE$

● サンプルプログラム

```
10 COLOR 15,1,1:SCREEN 4,3
20 SPRITE$(10)=STRING$(32,CHR$(&HFF))
30 FOR J=0 TO 15:READ C:C$=C$+CHR$(C):NEXT
40 COLOR SPRITE$(0)=C$
50 PUT SPRITE 0,(32,100),,10
60 I$=INPUT$(1)
70 SPRITE$(11)=STRING$(32,CHR$(&H3C))
80 COLOR SPRITE$(1)=STRING$(8,CHR$(&HC8))+
   STRING$(8,CHR$(&HC8))
90 PUT SPRITE 1,(32,100),,11
100 I$=INPUT$(1)
110 PUT SPRITE 0,(64,100)
120 I$=INPUT$(1)
130 DATA &H01,&H02,&H03,&H04,&H05,&H06,&H07,&H08
140 DATA &H08,&H07,&H06,&H05,&H04,&H03,&H02,&H01
```

CONT(コンティニュー)ステートメント

機能 停止されたプログラムを再開する

書式 CONT

文例 CONT

解説 [CTRL]+[STOP]の押下やSTOP文、END文によって停止されたプログラムの実行を再開します。

通常はデバッグのために用いられ、[CTRL]+[STOP]などで停止したプログラムの状態をLISTやPRINTなどで調べた後に再開することができます。ただし、停止中にプログラムの変更を行った場合、CONTによる再開はできません。

参照 STOP, END

COPY *1(コピー)ステートメント Disk

機能 ファイルをディスク間でコピーする

書式 COPY <転送元ファイルスペック> [TO <転送先ファイルスペック>]

文例 COPY "A:SAMPLE.DAT" TO "B:"

解説 <転送元ファイルスペック>で指定するファイルを<転送先ファイルスペック>へコピーします。<転送先ファイルスペック>を省略した場合、同じ名前でカレントドライブにコピーします。

例1：COPY "B:SAMPLE.DAT"
　BドライブのSAMPLE.DATを、カレントドライブへ名前を変えずにコピーします

例2：COPY "A:SAMPLE.DAT" TO "B:"
　AドライブのSAMPLE.DATを、Bドライブへ名前を変えずにコピーします

例3：COPY "SAMPLE.DAT" TO "SAMPLE.BAK"

カレントドライブのSAMPLE.DATのコピーを、同一ドライブ上に作成します

例4：COPY "A:SAMPLE.DAT" TO "B:SAMPLE.BAK"
Aドライブ のSAMPLE.DATを、SAMPLE.BAKの名前でBドライブへコピーします

COPY *2 (コピー)ステートメント　MSX2

機能 VRAM上の矩形領域をVRAM上の別の場所へコピーする

書式 COPY (<転送元左上座標>)-(<転送元右下座標>) [, <転送元ページ番号>] TO (<転送先座標>) [, <転送先ページ番号>] [, <論理演算子>]

文例 COPY (0,0)-(255,255) TO (0,0),1

解説 <転送元左上座標>、<転送元右下座標>で指定するVRAM上の矩形領域を、<転送先座標>で指定する場所へコピーします。SCREEN 5以上でのみ使用できます。

<転送元左上座標>、<転送元右下座標>、<転送先座標>には画面外の値を指定することができます。

<転送元ページ番号>は、転送元ページがアクティブページと同じ場合は省略できます。

<転送先ページ番号>は、転送先ページが<転送元ページ番号>と同じ場合は省略できます。

<論理演算子>に次の表の値を指定すると、転送元のカラーコード(S)と転送先のカラーコード(D)の論理演算を行います。省略すると、PSETが指定されたものとして動作します。

論理演算子	描画色
PSET	S
PRESET	not S
AND	S and D
OR	S or D
XOR	S xor D
TPSET	if S=0 then D else S
TPRESET	if S=0 then D else not S
TAND	if S=0 then D else S and D
TOR	if S=0 then D else S or D
TXOR	if S=0 then D else S xor D

※ S: 転送元のカラーコード
　D: 転送先のカラーコード

参照 SET PAGE, COPY *3, COPY *4

● サンプルプログラム

```
10 COLOR 15,1,1:SCREEN 5
20 CIRCLE (50,50),30
30 I$=INPUT$(1)
40 COPY (20,20)-(80,80) TO (100,100)
50 I$=INPUT$(1)
```

COPY *3 (コピー)ステートメント　MSX2

機能 VRAM-メモリ間でデータのコピーを行う

書式 COPY (<転送元左上座標>)-(<転送元右下座標>) [, <転送元ページ番号>] TO <配列変数名>
COPY <配列変数名> [, <転送方向>] TO (<転送先座標>) [, <転送先ページ番号>] [, <論理演算子>]

文例 COPY (0,0)-(31,31),1 TO GR
COPY GR TO (100,100),,XOR

解説 VRAM上の矩形領域と配列変数間でデータのコピーを行います。SCREEN 5以上でのみ使用できます。

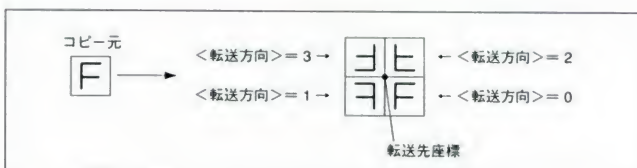
VRAMから<配列変数名>で指定する配列変数へコピーする場合は、DIM文で次の大きさ以上の領域を事前に確保しておく必要があります。

必要な領域の大きさ S = INT((X × Y + (K − 1)) / K + 4)
DIMで指定する添え字 D = INT((S + (A − 1)) / A − 1)

X: 転送領域の幅(ドット数)
Y: 転送領域の高さ(ドット数)
K: 1バイト当たりのドット数(SCREEN 5/7では2、SCREEN 6では4、SCREEN 8/10/11/12では1)
A: 配列変数の1要素のバイト数(配列変数が整数型の場合は2、単精度実数型は4、倍精度実数型は8)

<転送元ページ番号>、<転送先ページ番号>がアクティブページと同じ場合は省略できます。

<転送方向>には、データをVRAMに書き込む方向を0～3で指定します。省略すると、0が指定されたものとして動作します。

0: 左上から右下に向かってコピーします
1: 右上から左下に向かってコピーします
2: 左下から右上に向かってコピーします
3: 右下から左上に向かってコピーします

<論理演算子>についてはCOPY *2を参照してください。

参照 SET PAGE, COPY *2, COPY *4

● サンプルプログラム

```
10 COLOR 15,1,1:SCREEN 5
20 CIRCLE (50,50),10
35 DIM AR%[((21*21+(2-1))/2+4+(2-1))/2-1]
40 COPY (40,40)-(60,60) TO AR%
45 I$=INPUT$(1)
46 COPY AR% TO (100,100)
50 I$=INPUT$(1)
```

COPY *4 (コピー)ステートメント　MSX2 Disk

機能 VRAM-ディスク間またはメモリ-ディスク間でデータのコピーを行う

書式 COPY (<転送元左上座標>)-(<転送元右下座標>) [, <転送元ページ番号>] TO <ファイルスペック>
COPY <ファイルスペック> [, <転送方向>] TO (<転送先座標>) [, <転送先ページ番号>] [, <論理演算子>]
COPY <配列変数名> TO <ファイルスペック>
COPY <ファイルスペック> TO <配列変数名>

文例 COPY (0,0)-(255,211) TO "SAMPLE.DAT"

COPY "SAMPLE.DAT",1 TO (100,100),1
COPY GR TO "SAMPLE.DAT"
COPY "SAMPLE.DAT" TO GR

解説 <ファイルスペック>で指定するディスク上のファイルと、VRAM上の矩形領域または配列変数の間でデータのコピーを行います。メモリディスクやカセットテープのファイルに対しては使用できません。VRAM上の矩形領域に対してコピーを行えるのは、SCREEN 5以上だけです。

<転送元ページ番号>、<転送先ページ番号>がアクティブページと同じ場合は省略できます。

<配列変数名>で指定する配列変数の大きさと<転送方向>については、COPY[*3]を参照してください。

<論理演算子>については、COPY[*2]を参照してください。

参照 SET PAGE, COPY[*2], COPY[*3]

COS(コサイン)関数

機能 余弦(コサイン)を得る

書式 COS(<数式>)

文例 PRINT COS(3.14159266/4)＊2
実行結果　1.4142135601067

解説 <数式>の余弦(コサイン)を得ます。<数式>はラジアン単位で、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数です。

参照 SIN, TAN, ATN

CSNG(シー シングル)関数

機能 数値型の値を単精度型実数に変換し、その結果を得る

書式 CSNG(<数式>)

文例 PRINT CSNG(3.1415926535898)
実行結果　3.14159

解説 <数式>の値を単精度型実数に変換し、その結果を得ます。倍精度実数型の値は、有効数字6桁に丸められます。

変換結果が-9.99999E+62～9.99999E+62の範囲外になる場合は、エラー(Overflow)となります。

CSNGで得られる値は、数値型の値を単精度実数型変数に代入した場合と同じです。

参照 CDBL, CINT

CSRLIN(カーソルライン)関数

機能 現在のカーソル行位置を得る

書式 CSRLIN

文例 Y=CSRLIN

解説 関数を実行した時点のカーソルの行位置を得ます。画面上端の行を0とし、画面下へ向かうに従って大きくなる値を返します。

テキストモード(SCREEN 0～1)でのみ、意味のある値が得られます。

参照 POS, LOCATE

CVI(シー ブイ アイ)関数　Disk

機能 整数型の内部表現に対応した文字型の値を整数型に変換する

書式 CVI(<文字式>)

文例 I%=CVI(I$)

解説 <文字式>で指定する整数型の内部表現に対応した文字型の値を、整数型の値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。CVIは、MKI$が生成した2バイトの文字列を、整数型に変換します。

参照 OPEN, FIELD, GET, MKI$, MKS$, MKD$, CVS, CVD

CVS(シー ブイ エス)関数　Disk

機能 単精度実数型の内部表現に対応した文字型の値を単精度実数型に変換する

書式 CVS(<文字式>)

文例 S!=CVS(S$)

解説 文字式>で指定する単精度実数型の内部表現に対応した文字型の値を、単精度実数型の値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。CVSは、MKS$が生成した4バイトの文字列を、単精度実数型に変換します。

参照 OPEN, FIELD, GET, MKI$, MKS$, MKD$, CVI, CVD

CVD(シー ブイ ディー)関数　Disk

機能 倍精度実数型の内部表現に対応した文字型の値を倍精度実数型に変換する

書式 CVD(<文字式>)

文例 D#=CVD(D$)

解説 <文字式>で指定する倍精度実数型の内部表現に対応した文字型の値を、倍精度実数値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。CVDはMKD$で変換した8バイトの文字列を、倍精度実数型に変換します。

参照 OPEN, FIELD, GET, MKI$, MKS$, MKD$, CVI, CVS

DATA(データ)ステートメント

機能 READ文で読み出すデータを定義する

書式 DATA [<定数>] [, <定数> …]

文例 DATA 10,"ABC"

解説 READ文で読み出すデータの並びを定義します。DATA文を実行しても、何も行いません。

<定数>は、1つのDATA文中に複数書くことができます。また、1つのプログラム中に複数のDATA文を書くことができますが、1つのデータの並びとみなされます。

<定数>の型に制限はありませんが、定数式(1+2など)は許されません。また、READ文で指定する変数の型と、対応するDATA文の<定数>の型が一致するか、変換可能である必要があります。

<定数>の文字定数中に、カンマ(,)、コロン(:)を含む場合、引用符(")で囲んでください。

参照 READ, RESTORE

● サンプルプログラム

```
10 READ I%,F!,D#,S$
20 DATA 1,1.23456,3.1415926535898,abc
30 PRINT I%,F!,D#,S$
```

DEF FN(デファイン ファンクション)ステートメント

機能 関数を定義する

書式 DEF FN <関数名> [(<引数> [, <引数> …])]=<関数の定義式>

文例 DEF FN S(X,Y)=X^2+2＊X＊Y+Y^2

解説 関数を定義します。DEF FN文によって定義される関数の実行前に、DEF FNが実行されている必要があります。また、ダイレクトモードでの実行はできません。

<関数名>には定義する関数の名前を指定します。<関数名>の命名規則は、変数のそれに準じます(1～2文字の英数字)。

<引数>は<関数の定義式>内で使用する変数名に対応します。<関数の定義式>内で使用する変数名は、定義した関数を評価する際にのみ使われるため、プログラム中に同名の変数があっても影響を受けません(ローカル変数として扱われます)。

<引数>の型は、数値型、文字型、その混在のいずれも可能ですが、関数を呼び出すときに指定する実引数の型と一致しているか変換可能である必要があります。

また、<関数の定義式>は、数値型、文字型のいずれも可能ですが、関数の型と一致しているか変換可能である必要があります。

● サンプルプログラム

```
10 DEF FN PW(X,N)=X^N
20 FOR I=0 TO 16
30 PRINT FNPW(2,I)
40 NEXT
50 DEF FN CT$(X$,Y$,Z$)=X$+Y$+Z$
60 PRINT FN CT$("HELLO",", ","WORLD")
```

DEFINT(デファイン インテジャー)ステートメント

機能 変数を整数型として宣言する

書式 DEFINT <文字の範囲> [, <文字の範囲> …]

文例 DEFINT A,I-N

解説 <文字の範囲>で指定する文字で始まる変数の型を、整数型として宣言します。<文字の範囲>は英字1文字(例：A)か、英字1文字-英字1文字(例：X-Z)で指定します。

なお、DEFINTによる宣言より、型宣言文字によって指定される型が優先されます。また、型宣言が行われていない変数や、型宣言文字の指定がない変数は、すべて倍精度実数型とみなされます。

DEFINTによる型宣言は、CLEAR文の実行によって無効になりますので注意してください。

参照 DEFSNG, DEFDBL, DEFSTR, CLEAR

DEFDBL(デファイン ダブル)ステートメント

機能 変数を倍精度実数型として宣言する

書式 DEFDBL <文字の範囲> [, <文字の範囲> …]

文例 DEFDBL D,X-Z

解説 <文字の範囲>で指定する文字で始まる変数の型を、倍精度実数型として宣言します。詳しくはDEFINTの解説を参照してください。

参照 DEFINT, DEFSNG, DEFSTR, CLEAR

DEFSNG(デファイン シングル)ステートメント

機能 変数を単精度実数型として宣言する

書式 DEFSNG <文字の範囲> [, <文字の範囲> …]

文例 DEFSNG S,X-Z

解説 <文字の範囲>で指定する文字で始まる変数の型を、単精度実数型として宣言します。詳しくはDEFINTの解説を参照してください。

参照 DEFINT, DEFDBL, DEFSTR, CLEAR

DEFSTR(デファイン ストリング)ステートメント

機能 変数を文字型として宣言する

書式 DEFSTR <文字の範囲> [, <文字の範囲> …]

文例 DEFSTR D,X-Z

解説 <文字の範囲>で指定する文字で始まる変数の型を、文字型として宣言します。詳しくはDEFINTの解説を参照してください。

参照 DEFINT, DEFDBL, DEFSNG, CLEAR

DELETE(デリート)ステートメント

機能 プログラムの指定された行を削除する

書式 DELETE <行番号の範囲>

文例 DELETE 20-80

解説 プログラムから、<行番号の範囲>で指定する行を削除します。<行番号の範囲>の詳細は、LISTを参照してください。プログラム中でDELETE文を実行すると、指定行の削除後に実行を中断します。

参照 LIST, RENUM, NEW, MERGE

DIM(ディメンジョン)ステートメント

機能 配列変数を宣言し、メモリ領域に割り当てる

書式 DIM <変数名>(<添え字の最大値> [, <添え字の最大値> …]) [, <変数名>(<添え字の最大値> [, <添え字の最大値> …]) …]

文例 DIM A(10),B(8,8)

解説 配列変数を宣言し、メモリ上にその領域を割り当てます。<添え字の最大値>に指定できる値の最小は0(配列変数の要素数1)、最大はメモリの許す限りの値となります。DIMによる明示的な宣言を行わずに配列変数を用いる場合、<添え字の最大値>は自動的に10となります(配列変数の添え字は0～10で要素数は11)。また、4以上の次元を持つ配列変数を使用するには、DIMによる宣言が必要です(最大255次元まで指定できます)。

DIMで宣言された配列変数は、すべての要素が0(文字型の場合はヌルストリング)に初期化されます。

添え字を囲む括弧には、"()"以外に"[ ]"を使用することができます(例：DIM A[20]:A[20]=0)。

不要になった配列変数は、ERASE文で消去できます。宣言済みの配列変数は、ERASEかCLEARによって消去するまで宣言することはできません。

配列変数が使用するメモリの大きさは、次の式で計算できます。

配列変数の使用メモリ(単位:バイト) = 5 + A × N + 2 × D + 1 + S

A: 配列変数の1要素のバイト数(配列変数が整数型の場合は2、単精

度実数型は4、倍精度実数型は8、文字型は3)
N: 配列変数の要素数
D: 配列変数の次元(1～255)
S: 配列変数が数値型の場合は0、文字型の場合は各要素に含まれる文字列の総文字数

参照 ERASE, CLEAR

● サンプルプログラム

```
10 READ L
20 DIM S$[L-1]
30 FOR I=0 TO L-1:READ S$[I]:NEXT
40 R%=RND(-TIME)*L
50 PRINT S$[R%]
60 END
70 DATA 5
80 DATA ﾀﾞｲｷﾁ,ﾁｭｳｷﾁ,ｽｴｷﾁ,ｷｮｳ,ﾀﾞｲｷｮｳ
```

DRAW(ドロー)ステートメント

機能 画面に図形を描画する

書式 DRAW <文字式>

文例 DRAW "BM100,100E50L50F50L50U50E25F25D50"

解説 画面上の最終参照点(LP)を相対的に移動しながら、線を描きます。グラフィックモード(SCREEN 2～12)でのみ使用できます。

<文字式>には、**表2-1**のグラフィックマクロ命令を任意の数、記述します。

参照 SCREEN, COLOR, LINE, CIRCLE, PAINT

● サンプルプログラム

```
10 SCREEN 2
20 DRAW "BM0,0S8C15"
30 READ N:FOR I=0 TO N-1:READ D$:DRAW "XD$;":NEXT
40 FOR I=2 TO 8 STEP 2:DRAW "C=I;BM+8,+8S=I;
   R16D16L16U16":NEXT
50 I$=INPUT$(1)
60 DATA 3,"M+10,+20",E10F10G10H10,"BM100,
   100NU20ND20NR20NL20"
```

DSKF(ディスク フリー)関数 Disk

機能 ディスクの空き容量をクラスタ単位で得る

書式 DSKF(<ドライブ番号>)

グラフィックマクロ命令の種別	記号	意味	備考
移動マクロ命令	U <距離>	上へ移動する	<距離>はドット単位で指定し、これにスケールファクタを乗じた値が移動距離になる。ただし、斜め方向への移動と上下左右方向への移動では、同じ<距離>を指定しても移動距離が異なる(左図を参照)。 <距離>を省略した場合、1が指定されたとみなされる。
	D <距離>	下へ移動する	
	L <距離>	左へ移動する	
	R <距離>	右へ移動する	
	E <距離>	右上へ移動する	
	F <距離>	右下へ移動する	
	G <距離>	左下へ移動する	
	H <距離>	左上へ移動する	
	Mx,y	LPからx,yで指定する座標へ移動するxまたはyの前に符号(+か-)を付けると、LPからの相対移動を行う	xとyの一方を省略すると、指定された軸方向でのみ移動する(例："M,+10")。
	B	LPの移動のみを行い、描画しない	U、D、L、R、E、F、G、H、Mの直前に付加することで効果を発揮する(例："BM100,100")
	N	描画のみを行い、LPを移動しない	
回転マクロ命令	A <角度>	90°単位で回転する	U、D、L、R、E、F、G、Hと相対移動座標指定時のMで描画する図形を、回転して表示する。一度、指定すると、次のA命令まで有効である。<角度>は0～3の整数で表し、それぞれ、0°、90°、180°、270°に対応する。 A命令を指定しなかった場合、0が指定されたとみなされる。
色指定マクロ命令	C <カラーコード>	描画色を指定する	図形を描画するときの色を<カラーコード>で指定する<カラーコード>として使用できる値の範囲はスクリーンモードに依存する(COLORの解説を参照)。 C命令を指定しなかった場合、前景色が用いられる。
スケールファクタマクロ命令	S <整数>	スケールファクタを指定する	<整数>に指定した値の1/4がスケールファクタになる。U、D、L、R、E、F、G、Hと相対移動座標指定時のMで描画する図形を、拡大、縮小して表示する。一度指定すると、次のS命令まで有効である。 <整数>には1～255の値を指定する。S命令を指定しなかった場合、4が指定されたとみなされる。
ローカルマクロ命令	X <文字型変数>;	命令全体を<文字型変数>中のグラフィックマクロ命令に置き換える	例： D$="D50L50":DRAW "R50XD$;U50" DRAW "R50D50L50U50"と同じ
	= <数値型変数>;	命令全体を<数値型変数>の値を表現する文字列に置き換える	例： X=10:Y=5:L=20:DRAW "BM+=X;,-=Y;D=L;" DRAW "BM+10,-5D20"同じ

表2-1 グラフィックマクロ命令

文例 PRINT DSKF(0)

解説 <ドライブ番号>で指定するディスクの空き容量をクラスタ単位で得ます。クラスタの大きさは、ディスクのメディアによって異なります(下表を参照)。

メディア	クラスタサイズ
フロッピーディスク	1024バイト
ハードディスク	ハードディスク全体の容量によって異なる 1Kバイト、2Kバイト、4Kバイト、8Kバイト、16Kバイト、32Kバイト、64Kバイトのいずれかになる

END(エンド)ステートメント

機能 **プログラムを終了する**

書式 END

文例 END

解説 プログラムの実行を終了し、すべてのファイルを閉じた後に、コマンドレベルに戻ります。

END文は、プログラム中に複数置くことができます。プログラム最後のENDは省略できますが、その場合、ファイルは閉じられません。

参照 STOP

EOF(エンドオブファイル)関数

機能 **入力ファイルのデータをすべて読み出したかどうか調べ、結果を得る**

書式 EOF(<ファイル番号>)

文例 IF EOF(1) THEN END

解説 <ファイル番号>で指定する入力ファイルのデータを、すべて読み出したかどうか調べます。すべてのデータが読み出されており、これ以上のデータがない場合は真(-1)を、まだデータが残っている場合は偽(0)を返します。

<ファイル番号>で指定するファイルは、OPEN文のINPUTモードで開かれていなければなりません。

参照 OPEN, INPUT#

ERASE(イレース)ステートメント

機能 **配列変数を削除する**

書式 ERASE <配列変数名> [, <配列変数名> …]

文例 ERASE AR,VC$

解説 宣言済みの配列変数を削除し、割り当てられていたメモリを解放します。DIM文によって明示的に宣言された配列変数の他に、暗黙的に宣言された配列変数を削除することができます。

ERASE文で削除した配列変数は、明示的、暗黙的を問わず、再度宣言することができます。

参照 DIM

● **サンプルプログラム**

```
10 DIM AR(10),VC$(15)
20 TM(0)=1
30 ERASE AR,VC$,TM
40 DIM AR(20,15)
```

ERL(エラーライン)関数

機能 **エラーが発生した行の行番号を得る**

書式 ERL

文例 PRINT ERL

解説 最後にエラーが発生した行の行番号を得ます。エラーがダイレクトモードで発生した場合、ERLは65535を返します。

通常、ERLはON ERROR GOTOで指定したエラー処理ルーチンの中で使用します。なお、ERLを関係演算子と共に使用する場合、ERLを左辺に置くとRENUMで行番号を振りなおしたときに、比較対照の行番号も自動的に振りなおされます(例：IF ERL=1050 THEN RESUME NEXT)。

参照 ERR, ON ERROR GOTO, RESUME

ERR(エラー)関数

機能 **最後に発生したエラーのエラーコードを得る**

書式 ERR

文例 PRINT ERR

解説 最後に発生したエラーのエラーコードを得ます。通常、ERRはON ERROR GOTOで指定したエラー処理ルーチンの中で使用します。

参照 ERL, ON ERROR GOTO, RESUME, エラーコード表

ERROR(エラー)ステートメント

機能 **エラーを発生させる**

書式 ERROR <エラーコード>

文例 ERROR 15

実行結果　String too long

解説 <エラーコード>で指定するエラーを発生させます。<エラーコード>には、1～255の整数を指定します。

BASICがメッセージを設定していないエラー(エラーコード23, 26～49, 76～255)を発生させ、ON ERROR GOTOと組み合わせて独自のエラー処理を行うことができます。

参照 ERL, ERR, ON ERROR GOTO, RESUME, エラーコード表

EXP(エクスポネンシャル)関数

機能 **自然対数の底(e)のベキ乗を得る**

書式 EXP(<数式>)

文例 PRINT EXP(2.5)

実行結果　12.182493960702

解説 自然対数の底(e)の<数式>乗を得ます。

<数式>には、-149.66803104461499～145.06286085862の範囲で整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数になります。

FIELD(フィールド)ステートメント　Disk

機能 **ランダムアクセスファイルの入出力バッファに変数を割り当ててレコードを構成する**

書式 FIELD [#]<ファイル番号>, <フィールド幅> AS <文字変数名> [, <フィールド幅> AS <文字型変数名> …]

文例 FIELD #1,30 AS NM$,8 AS DY$,8 AS TM$

解説 <ファイル番号>で指定するランダムアクセスファイルの入出

カバッファに、<文字型変数名>で指定する文字型変数を割り当てます。入出力バッファはランダムアクセスファイルの1レコードに、文字型変数はレコードのフィールドに対応します。各フィールドの大きさは、<フィールド幅>に文字数単位で指定します。

レコードに複数のフィールドを指定するには、1つのFIELD文で複数の文字型変数を割り当てます。ただし、レコードに含まれるフィールドの<フィールド幅>の合計が、OPEN文で指定したレコード長を越えると、エラー(Filed overflow)となります。

FILED文は、実行するごとに入出力バッファの先頭から文字型変数を割り当てます。このとき、以前の割り当ても保持されるため、1個の入出力バッファに対して、複数のレコード形式を持つことができます。

FIELD文で割り当てた文字型変数を代入文の左辺や入力(INPUT等)に使用すると、以後、その変数をフィールドとして使用することはできなくなるので注意してください。これは、文字型変数の領域がランダムアクセスファイルの入出力バッファから文字領域に移るためです。

参照 OPEN, LSET, RSET, PUT, GET

FILES(ファイルズ)ステートメント　Disk

機能 **ディスク上に存在するファイルのファイル名を表示する**

書式 FILES [<ファイルスペック>]

文例 FILES "A:＊.BAS"

解説 <ファイルスペック>で指定するファイルのファイル名を表示します。<ファイルスペック>に一致するファイル名がディスク上に存在しない場合、エラー(File not found)となります。

<ファイルスペック>が省略された場合、カレントドライブのファイルをすべて表示します。また、<ファイルスペック>には、ワイルドカード(任意の文字または文字列に一致する特殊文字)を含むことができます。ワイルドカードとして使用できる文字については、次の表を参照してください。

ワイルドカード文字	意 味	例
?(疑問符)	任意の1文字に一致する	FILES "FILE?0.BAS" FILE10.BASやFILEX0.BASなどに一致する
＊(アスタリスク)	0文字以上の任意の文字列に一致する	FILES "TEST＊.＊" TEST.DATやTEST0001.ASCなどに一致する

参照 LFILES

FIX(フィックス)関数

機能 **数値の整数部を得る**

書式 FIX(<数式>)

文例 PRINT FIX(-3.14)

実行結果　-3

解説 <数式>の値の小数点以下を取り去った値を得ます。FIX(X)は、SGN(X) ＊ INT(ABS(X))と同じ結果になります。

<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数型になります。

参照 INT, CINT

FOR～NEXT(フォー～ネクスト)ステートメント

機能 **FOR文からNEXT文の間にある一連の文を、指定回数繰り返す**

書式 FOR <変数名>=<初期値> TO <終値> [STEP <増分>]

～

NEXT [<変数名> [, <変数名> …]]

文例 FOR I=0 TO 10 STEP 3:PRINT I;:NEXT I:PRINT ":"I

実行結果　0 3 6 9 : 12

解説 FOR文で指定した繰り返しの回数だけ、FOR文とNEXT文の間にある一連の文を繰り返します。<変数名>に指定した変数は繰り返し回数のカウンタとして使用され、最初に<初期値>に設定されます。そして、FOR文とNEXT文の間の文を実行した後、カウンタにSTEPで指定された<増分>を足し、カウンタの値が<終値>を越えていなければFOR文に戻り、越えていればNEXT文の次の文に移ります。なお、STEPを省略すると、<増分>は+1とみなされます。

FOR～NEXT文は入れ子にすることができますが、FOR文のカウンタに使用する<変数名>は異なるものにする必要があります。入れ子にしたFOR～NEXT文が同一点で終わる場合、NEXT文に複数の<変数名>を指定して、NEXT文を1つにまとめることができます。このときのNEXT文には、内側のFOR文で指定した<変数名>から、外側のFOR文で指定した<変数名>を順に並べます。また、NEXT文の<変数名>を省略した場合、最も近くのFOR文に対応しているとみなされます。

FRE(フリー)関数

機能 **未使用メモリの大きさを得る**

書式 FRE(<数式>)

FRE(<文字式>)

文例 PRINT FRE(0)

PRINT FRE("")

解説 引数に<数式>を指定すると、BASICが使用していないメモリ領域の大きさをバイト単位で得ます。

引数に<文字式>を指定した場合、文字領域中の不要な文字列を削除して使用可能領域を広げる作業(ガベージコレクション)を行った後、文字領域中の未使用領域の大きさをバイト単位で得ます。

引数に<数式>、<文字式>のいずれを指定する場合でも、その値は無視されます(引数の型のみが参照されます)。

GET(ゲット)ステートメント　Disk

機能 **ランダムアクセスファイルから入出力バッファ中へレコードを読み込む**

書式 GET [#]<ファイル番号> [, <レコード番号>]

文例 GET #1,RN

解説 <ファイル番号>で指定するランダムアクセスファイルから、<レコード番号>で指定するレコードを読み込みます。読み込んだレコードは入出力バッファに置かれ、FIELD文で指定した文字型変数を使って各フィールドにアクセスできます。<レコード番号>を省略すると、最後に実行したPUT文、GET文で使用したレコードの次のレコードから読み込みを行います。

読み込んだフィールドが文字型に変換された数値型である場合、CVI、CVS、CVDを使って数値型の値に戻すことができます。

参照 OPEN, FIELD, PUT, CVI, CVS, CVD

GET DATE(ゲット デート)ステートメント　MSX2

機能 日付を文字型変数に代入する

書式 GET DATE <文字型変数名> [, A]

文例 GET DATE DT$

解説 現在の日付を取得し、<文字型変数名>で指定する文字型変数に代入します。日付は、年月日が2桁ずつスラッシュ(/)で区切られた形式で代入されます(例：02/08/31)。

Aオプションを指定すると、設定されているアラームの日付が得られます。

参照 SET DATE, SET TIME, GET TIME

GET TIME(ゲット タイム)ステートメント　MSX2

機能 時刻を文字型変数に代入する

書式 GET TIME <文字型変数名> [, A]

文例 GET TIME TM$

解説 現在の時刻を取得し、<文字型変数名>で指定する文字型変数に代入します。時刻は、時分秒が2桁ずつコロン(:)で区切られた形式で代入されます(例：19:21:55)。

Aオプションを指定すると、設定されているアラームの時刻が得られます。

参照 SET TIME, SET DATE, GET DATE, TIME

GOSUB(ゴー サブ)ステートメント

機能 サブルーチンを呼び出す

書式 GOSUB <行番号>

文例 GOSUB 1000

解説 <行番号>で指定するサブルーチンを呼び出します。サブルーチン内でRETURN文が実行されると、GOSUBの次の文が実行されます。

サブルーチンは、GOSUBで指定する<行番号>からRETURN文までの一連の文のことです。1つのサブルーチン内に、複数のRETURN文を持つことができます。プログラム内の複数の場所で使う一連の文をサブルーチンとしてまとめておくと、GOSUB文によっていつでも実行することができます。

サブルーチンの中から他のサブルーチンを呼び出すこともできます(サブルーチンの多重化)。サブルーチンは、メモリのスタック領域の許す限り、多重化することができます。もし、スタック領域が足りなくなった場合、エラー(Out of memory)となります。なお、CLEAR文はスタック領域を初期化するため、サブルーチン内でCLEAR文を実行してはいけません。

サブルーチンは、必ずGOSUB文で実行する必要があります。もし、GOTO文などでサブルーチンを実行した場合、RETURN文の実行時にエラー(RETURN without GOSUB)となります。

参照 RETURN, ON GOSUB, GOTO, DEF FN

GOTO(ゴー トゥー)ステートメント

機能 指定した行へ制御を移す

書式 GOTO <行番号>
GO TO <行番号>

文例 GOTO 100

解説 <行番号>で指定する行へ制御を移します。

参照 ON GOTO, GOSUB

HEX$(ヘクサ ダラー)関数

機能 数値を16進表記の文字列に変換し、その結果を得る

書式 HEX$(<数式>)

文例 PRINT HEX$(100)
実行結果　64

解説 <数式>の値を16進表記の文字列に変換し、その結果を得ます。得られる値は文字型です。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、16進表記への変換は整数型への変換後に行われます。そのため、<数式>に指定できる値の範囲は-32768～32767(または0～65535)となります。

参照 VAL, BIN$, OCT$, STR$

IF～GOTO～ELSE(イフ～ゴー トゥー～エルス)ステートメント

機能 条件判断を行い、制御を指定行に移す

書式 IF <条件式> GOTO <行番号> [ELSE <行番号または文>]
IF <条件式> GO TO <行番号> [ELSE <行番号または文>]

文例 IF A=0 GOTO 100 ELSE A=A-1

解説 <条件式>が真(0以外)ならば<行番号>で指定する行に制御を移し、偽(0)ならばELSE文を実行します。ELSE文に行番号が指定された場合はその行に、文が指定された場合はその文の実行後、次の行に制御を移します。文には、複文(マルチステートメント)を指定することができます。ELSE文を省略した場合に<条件式>が偽になると、IF文の次の行が実行されます。

<条件式>には、関係演算式や論理演算式、算術演算式など数値型の値を指定する必要があります。

IF～GOTO～ELSE文は、ELSE文の中に別のIF文を置いて、入れ子にできます。

参照 IF～THEN～ELSE, GOTO

IF～THEN～ELSE(イフ～ゼン～エルス)ステートメント

機能 条件判断を行い、その結果に応じた操作を行う

書式 IF <条件式> THEN <行番号または文> [ELSE <行番号または文>]

文例 IF A=0 THEN END ELSE 100

解説 <条件式>が真(0以外)ならばTHEN文を、偽(0)ならばELSE文を実行します。THEN文、ELSE文に行番号が指定された場合はその行に、文が指定された場合はその文の実行後、次の行に制御を移します。文には、複文(マルチステートメント)を指定することができます。ELSE文を省略した場合に<条件式>が偽になると、IF文の次の行が実行されます。

<条件式>には、関係演算式や論理演算式、算術演算式など数値型の値を指定する必要があります。

IF～THEN～ELSE文は、THEN文、ELSE文の中に別のIF文を置いて、入れ子にできます。

参照 IF～GOTO～ELSE, GOTO

● サンプルプログラム

```
10 INPUT I
20 IF I >= 0 AND I < &H20 THEN PRINT BIN$(I):END
30 IF I < &H7F THEN PRINT CHR$(I) ELSE IF I <
   65536!THEN PRINT HEX$(I) ELSE 50
40 END
50 PRINT I
```

INKEY$(インキー ダラー)関数

機能 押されているキーを文字として得る

書式 INKEY$

文例 I$=INKEY$

解説 キーが押されている(キーバッファが空でない)場合、キーバッファの先頭1文字を取り出し、文字型の値として返します。キーが押されていない場合、ヌルストリング("")を返します。

INKEY$はINPUTやLINE INPUT、INPUT$と異なり、キーが押されるまで停止するということがありません。また、押されたキーは画面に表示されません。

INKEY$は、[CTRL]+[STOP]以外の文字すべてを読み取ります。

参照 INPUT, LINE INPUT, INPUT$

● サンプルプログラム

```
10 I$=INKEY$:IF I$<>"" THEN PRINT ASC(I$)
20 GOTO 10
```

INPUT(インプット)ステートメント

機能 キーボードから入力されるデータを変数に代入する

書式 INPUT ["<プロンプト文>";] <変数名> [, <変数名> …]

文例 INPUT "input name and age"; NM$,AG%

解説 <プロンプト文>で指定するメッセージを表示してユーザーに入力を促し、入力された値を<変数名>で指定する変数に代入します。

INPUT文を実行すると疑問符(?)が画面に表示され、キーボードからデータが入力されるのを待ちます。データが入力されるとその値が画面に表示され、<変数名>の変数に代入されます。<変数名>で指定する変数の型と入力したデータの型が一致しないか変換不能な場合は、「?Redo from start」を表示して、再度入力待ちとなります。

<変数名>をカンマ(,)区切りで複数指定した場合、入力するデータも<変数名>の数だけカンマで区切って入力します。入力データの個数が足りない場合は「??」を表示して再度入力待ちになり、個数が多い場合は「?Extra ignored」を表示して、余分のデータを無視します。

カンマや意味のある空白を含む文字列を文字型変数に代入する場合、文字列を引用符(")で囲んでください。この場合、文字列中に引用符を含めることはできません。

グラフィックモード(SCREEN 2～12)でINPUT文を実行すると、自動的にテキストモードに切り替わって入力待ちになります。

参照 LINE INPUT, INPUT$, INKEY$

INPUT#(インプット シャープ)ステートメント

機能 ファイルからデータを読み込み変数に代入する

書式 INPUT #<ファイル番号>, <変数名> [, <変数名> …]

文例 INPUT #1,I,S$

解説 <ファイル番号>で指定するファイルからデータを読み込み、<変数名>で指定する変数に代入します。ファイルは、OPEN文のINPUTモードで開かれている必要があります。

INPUT#で読み込むデータは、PRINT#で書き出したデータと対応しています。また、<変数名>の変数の型は、ファイルから読み込むデータの型と一致しているか変換可能である必要があります。

ファイル中、空白以外の最初の文字がデータの始まりとみなされます。データは、空白、改行、カンマ(,)のいずれかの文字によって区切られていなければなりません。

ファイル中のデータが文字型の場合、最初の文字が引用符(")であると、次の引用符までが1つのデータとみなされます。この場合、引用符を文字列中に含めることはできません。最初の文字が引用符以外の文字の場合、文字列はカンマか改行、255文字の読み込みによって区切られます。

参照 OPEN, CLOSE, PRINT#, LINE INPUT#, INPUT$, INPUT

INPUT$(インプット ダラー)関数

機能 指定されたファイルまたはキーボードから、指定された長さの文字列を得る

書式 INPUT$(<文字数> [, [#] <ファイル番号>])

文例 I$=INPUT$(10,#1)

解説 <ファイル番号>で指定するファイルから<文字数>分のデータを読み込み、その結果を得ます。得られる値は文字型です。ファイルは、OPEN文のINPUTモードで開かれている必要があります。<ファイル番号>を省略すると、キーボードから<文字数>分のデータが入力されるのを待ちます。その場合、入力した文字列は画面に表示されません。

INPUT$は[CTRL]+[STOP]以外の文字すべてを読み取るため、INPUTやLINE INPUTでは入力できない改行文字を取得することができます。

参照 OPEN, CLOSE, INPUT#, LINE INPUT#, INPUT, LINE INPUT, INKEY$

INSTR(インストリング)関数

機能 文字列中から指定の文字列を検索し、その位置を得る

書式 INSTR([<数式>,] <対象文字式>, <検索文字式>)

文例 PRINT INSTR("ABCDEF","BCD")

実行結果　2

解説 <対象文字式>から<検索文字式>を検索し、見つかれば発見した位置(<対象文字式>の先頭文字の位置を1とする値)を、見つからなければ0を得ます。

<数式>には、検索を始める位置を1～255の整数値で指定します。<数式>を省略すると、<対象文字列>の先頭から検索します。

INT(インテジャー)関数

機能 小数点以下を切り捨てた整数値を得る

書式 INT(<数式>)

文例 PRINT INT(3.14),INT(-3.14)

実行結果　3 -4

解説 <数式>の値を越えない最大の整数値を得ます。

<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数型になります。

参照 FIX, CINT

INTERVAL OFF(インターバルオフ)ステートメント

機能 タイマ割り込みを禁止する

書式 INTERVAL OFF

文例 INTERVAL OFF

解説 一定時間ごとに実行される割り込み処理の実行を、禁止します。

参照 INTERVAL ON, INTERVAL STOP, ON INTERVAL GOSUB

INTERVAL ON(インターバルオン)ステートメント

機能 タイマ割り込みを許可する

書式 INTERVAL ON

文例 INTERVAL ON

解説 一定時間ごとに実行される割り込み処理の実行を、許可します。

参照 INTERVAL OFF, INTERVAL STOP, ON INTERVAL GOSUB

INTERVAL STOP(インターバルストップ)ステートメント

機能 タイマ割り込みを保留する

書式 INTERVAL STOP

文例 INTERVAL STOP

解説 一定時間ごとに実行される割り込み処理の実行を、保留します。保留された割り込み処理は、INTERVAL ONが実行されたときに実行されます。なお、INTERVAL STOPは、INTERVAL ON状態で実行した場合にのみ有効です。

参照 INTERVAL ON, INTERVAL OFF, ON INTERVAL GOSUB

KEY(キー)ステートメント

機能 ファンクションキーの内容を設定する

書式 KEY <キー番号>, <文字式>

文例 KEY 5,CHR$(12)+"RUN"+CHR$(13)

解説 <キー番号>で指定するファンクションキーの内容に、<文字式>で表される文字列を設定します。設定した内容は、KEY LISTで参照できます。

<キー番号>は1～10の整数で、それぞれ[F1]～[F10]に対応しています。<文字式>は最大15文字の文字列です。文字列中に制御文字(コントロールコード)を含める場合は、CHR$を連結演算子でつないでください。

一度設定したファンクションキーの内容は、設定し直すかシステムをリセットするまで変化しません。

参照 KEY LIST, KEY ON [*1], KEY OFF [*1]

KEY LIST(キーリスト)ステートメント

機能 ファンクションキーの内容を画面に表示する

書式 KEY LIST

文例 KEY LIST

解説 [F1]～[F10]のすべてのファンクションキーの内容を、画面に表示します。起動直後には、次の内容が設定されています。

[F1] "color"
[F2] "auto"
[F3] "goto"
[F4] "list"
[F5] "run"+CHR$(13)
[F6] "color 15,4,7"+CHR$(13)
[F7] "cload"+chr$(34) (MSX2+以降では"load"+chr$(34))
[F8] "cont"+CHR$(13)
[F9] "list."+CHR$(13)+CHR$(31)+CHR$(31)
[F10] CHR$(12)+"run"+CHR$(13)

参照 KEY, KEY ON [*1], KEY OFF [*1]

KEY OFF [*1] (キーオフ)ステートメント

機能 画面最下行のファンクションキー表示設定を取り消す

書式 KEY OFF

文例 KEY OFF

解説 KEY ON [*1]によるファンクションキーの表示設定を取り消します。

参照 KEY ON [*1], KEY LIST, KEY

KEY OFF [*2] (キーオフ)ステートメント

機能 ファンクションキーの押下に対応する割り込み処理を禁止する

書式 KEY(<キー番号>) OFF

文例 KEY(1) OFF

解説 ファンクションキーが押されたときに実行される割り込み処理の実行を、禁止します。<キー番号>には1～10を整数を指定します([F1]～[F10]に対応)。

参照 KEY ON [*2], KEY STOP, ON KEY GOSUB

KEY ON [*1] (キーオン)ステートメント

機能 画面最下行にファンクションキーの内容を表示する

書式 KEY ON

文例 KEY ON

解説 ファンクションキーの内容を画面最下行に表示する設定に変更します。通常は[F1]～[F5]が表示され、[SHIFT]の押下中は[F6]～[F10]が表示されます。

参照 KEY OFF [*1], KEY LIST, KEY

KEY ON [*2] (キーオン)ステートメント

機能 ファンクションキーの押下に対応する割り込み処理を許可する

書式 KEY(<キー番号>) ON

文例 KEY(5) ON

解説 ファンクションキーが押されたときに実行される割り込み処理の実行を、許可します。<キー番号>には1～10を整数を指定します([F1]～[F10]に対応)。

参照 KEY OFF [*2], KEY STOP, ON KEY GOSUB

KEY STOP(キーストップ)ステートメント

機能 ファンクションキーの押下に対応する割り込み処理を保留する

書式 KEY(<キー番号>) STOP

文例 KEY(10) STOP

解説 ファンクションキーが押されたときに実行される割り込み処理の実行を、保留します。<キー番号>には1～10の整数を指定します([F1]～[F10]に対応)。保留された割り込み処理は、KEY ON [*2]が実行されたときに実行されます。なお、KEY STOP [*2]は、KEY ON [*2]状態で実行した場合にのみ有効です。

参照 KEY ON [*2], KEY OFF [*2], ON KEY GOSUB

KILL(キル)ステートメント Disk

機能 ディスク上のファイルを消去する

書式 KILL <ファイルスペック>

文例 KILL "TEMP.DAT"

解説 <ファイルスペック>で指定するディスク上のファイルを消去します。<ファイルスペック>にワイルドカードを使用して、複数のファイルを一度に消去できます。既に開かれているファイルに対してKILL文を実行すると、エラー(File still open)となります。

参照 FILES

LEFT$(レフトダラー)関数

機能 文字列の左側から、指定された長さの文字列を得る

書式 LEFT$(<文字式>, <数式>)

文例 PRINT LEFT$("ABCDEFG",3)

実行結果 ABC

解説 <文字式>で表される文字列の左側(先頭)から、<数式>の値の文字数分の文字列を得ます。得られる値は文字型です。<数式>に指定できる値の範囲値は、0～255です。

<文字式>の総文字数より<数式>の値が大きい場合、<文字式>全体を得ます。また、<数式>の値が0の場合、常にヌルストリング("")を得ます。

グラフィックキャラクタは、1文字当たり2文字として数えられます。

参照 RIGHT$, MID$ [*1]

LEN(レングス)関数

機能 文字列の総文字数を得る

書式 LEN(<文字式>)

文例 PRINT LEN("ABC")

実行結果 3

解説 <文字式>で表される文字列の総文字数を得ます。制御文字(コントロールコード)や空白も含めて、文字数を数えます。

グラフィックキャラクタは、1文字当たり2文字として数えられます。

LET(レット)ステートメント

機能 変数に値を代入する

書式 [LET] <変数名>=<式>

文例 A#=3.1415926535

解説 <変数名>で指定する変数に、<式>が持つ値を代入します。LETは省略できます。

<式>の型は、数値型、文字型のいずれも可能ですが、<変数名>で指定する変数の型と一致しているか変換可能である必要があります。

LINE(ライン)ステートメント

機能 線または矩形を描く

書式 LINE [[STEP] (x1, y1)] - [STEP] (x2, y2) [, <カラーコード>] [, B [F]] [, <論理演算子>]

文例 LINE (100,100)-(200,200),10,BF

解説 (x1, y1)と(x2, y2)で表される2点を結ぶ直線を描きます。グラフィックモード(SCREEN 2～12)でのみ使用できます。(x1, y1)と(x2, y2)に指定可能な値については、CIRCLEを参照してください。

(x1, y1)を省略すると、最終参照点(LP)が指定されたものとみなされます。(x1, y1)を最終参照点(LP)からの相対座標で指定する場合、(x1, y1)の前にSTEPを付加します。また、(x2, y2)を(x1, y1)からの相対座標で指定する場合、(x2, y2)の前にSTEPを付加します。LPはLINEの実行後、(x2, y2)に設定されます。

<カラーコード>を指定すると、指定した色で直線が描かれます。<カラーコード>を省略した場合は、COLOR文で指定した前景色が用いられます。

Bを指定すると、(x1, y1)と(x2, y2)で表される2点を対角とする矩形を描きます(Box)。また、BFを指定すると、(x1, y1)と(x2, y2)で表される2点を対角とする矩形を描き、その内側を塗りつぶします(Box Fill)。

<論理演算子>を指定すると、<カラーコード>を転送元のカラーコードとする論理演算を行います。論理演算子は、SCREEN 5～12でのみ使用できます。論理演算子の詳細は、COPY [*2]を参照してください。

参照 COLOR, SCREEN, CIRCLE, PSET, PRESET, PAINT, COPY

● サンプルプログラム

```
10 SCREEN 2
20 LINE (10,10)-STEP (100,100),15
30 LINE -STEP (-50,0),12
40 LINE STEP (110,-10)-STEP (10,50),10,B
50 LINE (90,50)-STEP (30,25),7,BF
60 FOR I=0 TO 3
70 LINE STEP (-10,-10)-STEP (20,20),3,B
80 NEXT
90 I$=INPUT$(1)
```

LINE INPUT(ライン インプット)ステートメント

機能 キーボードから入力される1行のデータを文字型変数に代入する

書式 LINE INPUT ["<プロンプト文>";] <文字型変数名>

文例 LINE INPUT "input name"; NM$

解説 <プロンプト文>で指定するメッセージを表示してユーザーに入力を促し、[RETURN]が入力されるまでのすべてのデータを<文字型変数名>で指定する文字型変数に代入します。キーボードから入力できる文字列の長さは、最大255文字です。

INPUT文と異なり、LINE INPUT文は疑問符を表示しません。また、複数の変数に同時に値を代入することもできません。

参照 LINE INPUT#, INPUT, INPUT$, INKEY$

LINE INPUT#(ライン インプット シャープ)ステートメント

機能 ファイルから1行分のデータを読み込み、文字型変数に代入する

書式 LINE INPUT # <ファイル番号>, <文字型変数名>

文例 INPUT #1,I,S$

解説 <ファイル番号>で指定するファイルから1行分(255文字以内)のデータを読み込み、<文字型変数名>で指定する文字型変数に代入します。ファイルは、OPEN文のINPUTモードで開かれている必要があります。

LINE INPUT#文は、改行までを1行とみなして読み込みますが、改行文字が文字型変数に代入されることはありません。ただし、1行の長さが255文字を越える場合、255文字までが読み込まれ、行の残りは次のLINE INPUT#文で読み込まれます。

参照 OPEN, CLOSE, PRINT#, INPUT#, INPUT$, LINE INPUT

LIST(リスト)ステートメント

機能 メモリ上のプログラムを画面に表示する

書式 LIST [<行番号の範囲>]

文例 LIST 20-80

解説 プログラムの<行番号の範囲>で指定する行を表示します。<行番号の範囲>の範囲を省略すると、プログラム全体を表示します。<行番号の範囲>には、次の表のような指定を行えます。

指定形式	意 味	例
<行番号>	指定された1行を表示する	LIST 100
<始点行番号>-<終点行番号>	<始点行番号>から<終点行番号>までを表示する	LIST 30-90
-<終点行番号>	先頭行から<終点行番号>までを表示する	LIST -50

BASICが現在着目している(スクリーンエディタやLIST文などで最後に参照した)行番号として、ピリオド(.)を使用できます(例：LIST 50-.)。

プログラムを表示している途中で[STOP]を押すと表示は中断し、再度[STOP]を押すと再開します。また、表示の途中に[CTRL]+[STOP]を押すと、表示が終了します。

プログラム中でLIST文を実行すると、指定行の表示後に実行を中断します。

LOAD(ロード)ステートメント

機能 ファイルに保存されたプログラムをメモリ上に読み込む

書式 LOAD <ファイルスペック> [, R]

文例 LOAD "TEST.BAS"

解説 <ファイルスペック>で指定するファイルを、BASICプログラムとしてメモリ上に読み込みます(ロードします)。LOAD文を実行すると、LOAD文実行前にメモリ上にあったプログラムは消去されます。また、開かれているすべてのファイルが閉じられ、変数の値が初期化されます。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブのファイルを読み出します(Disk BASICが存在する場合)。

Rオプションを指定すると、プログラムを読み込んだ後、すぐに実行を開始します。

LOAD文は指定されたファイルを見つけてプログラムのロードを始めるまでは、メモリ上のプログラムを消去しません。

参照 SAVE, RUN [*2], MERGE

LOC(エルオーシー)関数 Disk

機能 ファイル中の現在位置を得る

書式 LOC(<ファイル番号>)

文例 IF LOC(1) > 100 THEN END

解説 <ファイル番号>で指定するファイルがランダムアクセスファイルの場合、最後に入出力(PUT, GET)を行ったレコードのレコード番号を得ます。ファイルを開いた直後は0を得ます。

<ファイル番号>で指定するファイルがシーケンシャルファイルの場合、そのファイルが開かれてから読み書きされた量をバイト単位で得ます。BASICはファイルを256バイト単位で読み書きするため、LOCで得られる値も256バイト区切りになります。

ファイルをINPUTモードで開いた場合、開いた直後は256(ファイルが256バイト未満の場合はそのファイルの大きさ)を得ます。これは、ファイルを開いたときにBASICが最初の256バイトを読み込むためです。

参照 OPEN, FIELD, PUT, GET

● サンプルプログラム

```
10 OPEN "temp.dat" FOR OUTPUT AS #1
20 FOR I=0 TO 100:PRINT I;LOC(1):PRINT #1,I:NEXT
30 CLOSE #1
40 OPEN "temp.dat" FOR INPUT AS #1
50 FOR I=0 TO 100:PRINT LOC(1);:INPUT #1,V:PRINT V
60 NEXT
70 CLOSE #1
80 OPEN "temp.dat" AS #1 LEN=100
90 FIELD #1,10 AS A$,20 AS B$,30 AS C$,40 AS D$
100 LSET A$="MSX1":RSET B$="MSX2":LSET C$="MSX2+":
    RSET D$="MSXturboR"
110 PUT #1:CLOSE #1
120 OPEN "temp.dat" AS #1 LEN=100
130 FIELD #1,10 AS A$,20 AS B$,30 AS C$,40 AS D$
140 PRINT LOC(1):GET #1:PRINT LOC(1)
150 PRINT A$:PRINT B$:PRINT C$:PRINT D$
160 CLOSE #1
```

LOCATE(ロケート)ステートメント

機能 カーソルを指定された位置に移動する

書式 LOCATE [<桁位置>] [, <行位置>] [, <カーソルスイッチ>]

文例 LOCATE 10,10

解説 カーソルを<桁位置>、<行位置>で指定する位置に移動します。テキストモード(SCREEN 0～1)でのみ、有効です。

<桁位置>には、画面左端の桁を0とし、画面右へ向かうに従って大きくなる値を整数で指定します。省略した場合は、0が指定されたものとみなされます。

<行位置>には、画面上端の桁を0とし、画面下へ向かうに従って大きくなる値を、整数で指定します。省略した場合は、LOCATE文を実行した時の行位置が指定されたものとみなされます。

<カーソルスイッチ>には、カーソルの表示状態を設定します。<カーソルスイッチ>に0を指定すると、入力待ちのときのみカーソルが表示され、0以外の値を指定すると、常にカーソルが表示されます。省

略した場合は、現在の設定が適用されます。

参照 POS, CSRLIN

● **サンプルプログラム**

```
10 CLS
20 FOR I=0 TO 20
30 LOCATE I,I
40 PRINT "*"
50 NEXT
```

LOF(エル オー エフ)関数　Disk

機能 **ファイルの大きさを得る**

書式 LOF(<ファイル番号>)

文例 FOR I=0 TO LOF(1)/256

解説 <ファイル番号>で指定するファイルの大きさをバイト単位で得ます。

LOG(ログ)関数

機能 **自然対数を得る**

書式 LOG(<数式>)

文例 PRINT LOG(10)
実行結果　2.302585092994

解説 <数式>の自然対数を得ます。<数式>には、0より大きい整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数になります。

LSET(レフト セット)ステートメント　Disk

機能 **ランダムアクセスファイルの入出力バッファに、データを左詰で転送する**

書式 LSET <文字型変数名>=<文字式>

文例 LSET NM$="MSX"

解説 <文字型変数名>で指定するランダムアクセスファイルのフィールドに、<文字式>で表される文字列を転送します。転送はランダムアクセスファイルの入出力バッファに対して行われ、実際にファイルに出力されるのはPUT文の実行時です。

<文字式>の文字列の長さがFIELD文で指定したフィールド幅よりも小さい場合、左詰で転送され、余った部分は空白で埋められます。文字列の長さがフィールド幅よりも大きい場合は、文字列の右側が切り捨てられます。

LSETは文字型のデータしか扱えないため、数値型のデータをフィールドに使用する場合はMKI$、MKS$、MKD$を使って文字型に変換する必要があります。

参照 OPEN, FIELD, PUT, RSET, MKI$, MKS$, MKD$

MAX FILES(マックス ファイルズ)ステートメント

機能 **使用するファイルの数を設定する**

書式 MAX FILES=<ファイル数>

文例 MAX FILES=3

解説 OPEN文で同時に開くファイルの数を設定します。<ファイル数>は0〜15の値を持つ整数です。ファイル1個につき、267バイトのメモリを消費するため、必要なファイル数だけを指定してください。

MAX FILES文を実行すると、すべての変数が初期化され、開かれているすべてのファイルが閉じられます。

起動直後は、MAX FILES=1となっています。

参照 OPEN, CLOSE

MERGE(マージ)ステートメント

機能 **指定したアスキー形式ファイルのプログラムを、メモリ上のプログラムと混合(マージ)する**

書式 MERGE <ファイルスペック>

文例 MERGE "ADD.BAS"

解説 <ファイルスペック>で指定するファイルのプログラムとメモリ上のプログラムを混合し、1つのプログラムにしてメモリ上に置きます。<ファイルスペック>で指定するファイルは、アスキー形式で保存されていなければなりません。ファイル中のプログラムとメモリ上のプログラムに同一の行番号があった場合、ファイル中の行で置き換えます。

MERGE文は、実行を完了するとコマンドレベルに戻ります。

参照 SAVE, LOAD

MID$ *1(ミドル ダラー)関数

機能 **文字列の指定した位置から、指定された長さの文字列を得る**

書式 MID$(<文字式>, <位置> [, <文字数>])

文例 PRINT MID$("ABCDEFG",2,3)
実行結果　BCD

解説 <文字式>で表される文字列の<位置>以降から、<文字数>分の文字列を得ます。得られる値は文字型です。<位置>、<文字数>に指定できる値の範囲は、それぞれ1〜255、0〜255です。

<文字数>を省略した場合や<文字式>の<位置>以降の文字数より<文字数>の値が大きい場合、<位置>以降の文字列すべてを得ます。また、<文字数>の値が0の場合や<文字式>の総文字数が<位置>より小さい場合、ヌルストリング("")を得ます。

グラフィックキャラクタは、1文字当たり2文字として数えられます。

参照 MID$ *2, LEFT$, RIGHT$

MID$ *2(ミドル ダラー)ステートメント

機能 **文字変数の一部を他の文字列で置き換える**

書式 MID$(<文字変数名>, <位置> [, <文字数>])=<文字式>

文例 A$="ABCDEFGH":MID$(A$,2,3)="3456":PRINT A$
実行結果　A345EFGH

解説 <文字変数名>で指定する文字変数の<位置>以降の<文字数>分の文字列を、<文字式>の先頭から<文字数>分の文字列で置き換えます。<位置>、<文字数>に指定できる値の範囲は、それぞれ1〜255、0〜255です。

<文字数>を省略した場合や<文字式>の長さより<文字数>の値が大きい場合、<文字式>のすべての文字と置き換えます。

<文字変数名>の文字変数の長さより<位置>の値が大きい場合、エラー(Illegal function call)となります。

グラフィックキャラクタは、1文字当たり2文字として数えられます。

参照 MID$ *1

MKI$(エムケーアイダラー)関数 Disk

機能 整数型の値を内部表現に対応した文字型に変換する

書式 MKI$(<数式>)

文例 LSET NM$=MKI$(I%)

解説 <数式>で指定する整数型の値を、BASICの内部表現に対応した文字型の値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。MKI$は整数型の値を2バイトの文字列に変換します。MKI$で変換した文字型の値を整数型に戻すには、CVIを使用します。

参照 OPEN, FIELD, LSET, RSET, PUT, MKS$, MKD$, CVI, CVS, CVD

MKS$(エムケーエスダラー)関数 Disk

機能 単精度実数型の値を内部表現に対応した文字型に変換する

書式 MKS$(<数式>)

文例 LSET NM$=MKS$(S!)

解説 <数式>で指定する単精度実数型の値を、BASICの内部表現に対応した文字型の値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。MKS$は単精度実数型の値を4バイトの文字列に変換します。MKS$で変換した文字型の値を単精度実数型に戻すには、CVSを使用します。

参照 OPEN, FIELD, LSET, RSET, PUT, MKI$, MKD$, CVI, CVS, CVD

MKD$(エムケーディーダラー)関数 Disk

機能 倍精度実数型の値を内部表現に対応した文字型に変換する

書式 MKD$(<数式>)

文例 LSET NM$=MKD$(D#)

解説 <数式>で指定する倍精度実数型の値を、BASICの内部表現に対応した文字型の値に変換します。ランダムアクセスファイルは文字型しか扱えないため、数値型は書き込み時に文字型に変換(MKI$, MKS$, MKD$)し、読み込み時に数値型に戻す(CVI, CVS, CVD)必要があります。MKD$は倍精度実数型の値を8バイトの文字列に変換します。MKD$で変換した文字型の値を倍精度実数型に戻すには、CVDを使用します。

参照 OPEN, FIELD, LSET, RSET, PUT, MKI$, MKS$, CVI, CVS, CVD

NAME(ネーム)ステートメント Disk

機能 ディスク上のファイル名を変更する

書式 NAME <ファイルスペック> AS <ファイル名>

文例 NAME "*.ASC" AS "*.BAS"

解説 <ファイルスペック>で指定するディスク上のファイル(群)の名前を、<ファイル名>で指定する名前に変更します。<ファイルスペック>で指定するファイルは存在してなければならず、<ファイル名>で指定するファイルは存在していてはなりません。NAME文はファイル名を変更しますが、ファイルの内容や大きさを変更することはありません。

<ファイルスペック>、<ファイル名>にはワイルドカード(FILESを参照)が使えます。<ファイルスペック>でワイルドカードを使用すると、FILESと同じようにディスク上のファイル名に一致します。また、<ファイル名>に使用すると、使用した部分を変更前のファイル名と同じくすることができます。

参照 FILES

NEW(ニュー)ステートメント

機能 メモリ上のプログラムを消去し、変数を初期化する

書式 NEW

文例 NEW

解説 メモリ上のプログラムをすべて消去し、変数を初期化します。また、開かれているファイルをすべて閉じます。

NEW文は、実行を完了するとコマンドレベルに戻ります。

参照 DELETE, ERASE

OCT$(オクタルダラー)関数

機能 数値を8進表記の文字列に変換し、結果を得る

書式 OCT$(<数式>)

文例 PRINT OCT$(100)
実行結果　144

解説 <数式>の値を8進表記の文字列に変換し、その結果を得ます。得られる値は文字型です。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、8進表記への変換は整数型への変換後に行われます。そのため、<数式>に指定できる値の範囲は-32768～32767(または0～65535)となります。

参照 VAL, BIN$, HEX$, STR$

ON ERROR GOTO(オンエラーゴートゥー)ステートメント

機能 エラー処理ルーチンの開始行を定義する

書式 ON ERROR GOTO <行番号>

文例 ON ERROR GOTO 1000

解説 エラーが発生したときに呼び出されるエラー処理ルーチンの開始行を、<行番号>で定義します。<行番号>に指定した行が存在しないと、エラー(Undefined line number in XXXXX)となります。

エラーが発生すると、通常はBASICがエラーメッセージを表示してプログラムを終了し、コマンドレベルに戻ります。しかし、ON ERROR GOTOでエラー処理ルーチンを定義しておくと、BASICは通常のエラー処理を行わず、エラー処理ルーチンに制御を移します。エラー処理ルーチンでは任意のエラー処理を行い、RESUME文でエラーからの回復を行います。

<行番号>に0を指定(ON ERROR GOTO 0)すると、エラー処理ルーチンの定義は無効になり、以後、BASICによる通常のエラー処理が行われます。また、エラー処理ルーチンでON ERROR GOTO 0を実行すると、通常のエラー処理を行います。

ON ERROR GOTOによるエラー処理ルーチンの定義は、別のON ERROR GOTO文かRUN文、CLEAR文を実行するまで有効です。

参照 RESUME, ERL, ERR, ERROR

● サンプルプログラム

```
10 ON ERROR GOTO 80
20 INPUT "file name:";N$
30 OPEN N$ FOR INPUT AS #1
40 IF EOF(1) THEN END
50 LINE INPUT #1,L$
60 PRINT L$
70 GOTO 40
80 IF ERR=62 THEN PRINT "bad drive name":RESUME 10
90 IF ERR=56 THEN PRINT "bad file name": RESUME 10
100 IF ERR=70 THEN PRINT "set disette":RESUME 10
110 IF ERR=53 THEN PRINT "file not found":RESUME 10
120 ON ERROR GOTO 0
```

ON GOTO(オン ゴー トゥー)ステートメント

機能 **指定した行のいずれかへ制御を移す**

書式 ON <数式> GOTO <行番号> [, <行番号> …]
ON <数式> GO TO <行番号> [, <行番号> …]

文例 ON SW GOTO 100,200,300

解説 <数式>の値に応じて、指定した<行番号>のいずれかに制御を移します。<行番号>の並びは、左から順に1から始まる整数に対応しています(例：<数式>の値が3であれば、左から3番目の<行番号>に制御を移す)。

<数式>の値が行番号の数より大きいか0の場合、ON GOTOの次の文に制御が移ります。また、<数式>の値が負の場合、エラー(Illegal function call)となります。

参照 GOTO, ON GOSUB

ON GOSUB(オン ゴー サブ)ステートメント

機能 **指定したサブルーチンのいずれかを呼び出す**

書式 ON <数式> GOSUB <行番号> [, <行番号> …]

文例 ON SW GOSUB 100,200,300

解説 <数式>の値に応じて、<行番号>で指定するサブルーチンを呼び出します。<行番号>の並びは、左から順に1から始まる整数に対応しています(例：<数式>の値が3であれば、左から3番目の<行番号>のサブルーチンを呼び出す)。

<数式>の値が行番号の数より大きいか0の場合、ON GOSUBの次の文に制御が移ります。また、<数式>の値が負の場合、エラー(Illegal function call)となります。

参照 GOSUB, ON GOTO, RETURN

ON INTERVAL GOSUB(オン インターバル ゴー サブ)ステートメント

機能 **一定時間ごとに実行されるサブルーチンの開始行を定義する**

書式 ON INTERVAL=<時間> GOSUB <行番号>

文例 ON INTERVAL=60 GOSUB 1000

解説 <時間>で指定する時間ごとに実行されるサブルーチンの開始行を、<行番号>で定義します。<時間>には、1/60秒単位の整数(1～65535)を指定します。

指定したサブルーチンは、INTERVAL ON(割り込み許可)状態で指定した時間が経過したときに、通常処理を中断する割り込み処理として実行されます。サブルーチンの実行中はINTERVAL STOP(割り込み保留)状態になり、RETURN文によって割り込みを終了して通常処理に復帰するとINTERVAL ON状態になります(二重割り込みの防止)。割り込みを禁止する場合は、INTERVAL OFFを実行してください。

参照 INTERVAL ON, INTERVAL OFF, INTERVAL STOP, ON GOSUB, TIME

ON KEY GOSUB(オン キー ゴー サブ)ステートメント

機能 **ファンクションキーが押されたときに実行されるサブルーチンの開始行を定義する**

書式 ON KEY GOSUB [<行番号>] [, <行番号> …]

文例 ON KEY GOSUB 1000, 2000,,4000

解説 ファンクションキーが押されたときに実行されるサブルーチンの開始行を、<行番号>で定義します。行番号は最大10個まで並べることが可能で、左から[F1]～[F10]に対応しています。

指定したサブルーチンは、KEY ON *2 (割り込み許可)状態でファンクションキーが押されたときに、通常処理を中断する割り込み処理として実行されます。サブルーチンの実行中はKEY STOP(割り込み保留)状態になり、RETURN文によって割り込みを終了して通常処理に復帰するとKEY ON *2状態になります(二重割り込みの防止)。割り込みを禁止する場合は、KEY OFF *2を実行してください。

参照 KEY ON *2, KEY OFF *2, KEY STOP, ON GOSUB

ON SPRITE GOSUB(オン スプライト ゴー サブ)ステートメント

機能 **スプライトが重なったときに実行されるサブルーチンの開始行を定義する**

書式 ON SPRITE GOSUB <行番号>

文例 ON SPRITE GOSUB 1000

解説 スプライトが重なったときに実行されるサブルーチンの開始行を、<行番号>で定義します。

指定したサブルーチンは、SPRITE ON(割り込み許可)状態でスプライトが重なったときに、通常処理を中断する割り込み処理として実行されます。サブルーチンの実行中はSPRITE STOP(割り込み保留)状態になり、RETURN文によって割り込みを終了して通常処理に復帰するとSPRITE ON状態になります(二重割り込みの防止)。割り込みを禁止する場合は、SPRITE OFFを実行してください。

参照 PUT SPRITE, SPRITE ON, SPRITE OFF, SPRITE STOP, ON GOSUB

ON STOP GOSUB(オン ストップ ゴー サブ)ステートメント

機能 **[CTRL]+[STOP]が押されたときに実行されるサブルーチンの開始行を定義する**

書式 ON STOP GOSUB <行番号>

文例 ON STOP GOSUB 1000

解説 [CTRL]+[STOP]が押されたときに実行されるサブルーチンの開始行を、<行番号>で定義します。

指定したサブルーチンは、STOP ON(割り込み許可)状態で[CTRL]+[STOP]が押されたときに、通常処理を中断する割り込み処理として実行されます。サブルーチンの実行中はSTOP STOP(割り込み保留)状態になり、RETURN文によって割り込みを終了して通常処理に復帰するとSTOP ON状態になります(二重割り込みの防止)。割り込みを禁止する場合は、STOP OFFを実行してください。

参照 STOP ON, STOP OFF, STOP STOP, ON GOSUB

ON STRIG GOSUB(オンエストリガゴーサブ)ステートメント

機能 **スペースキーやジョイスティックのトリガが押されたときに実行されるサブルーチンの開始行を定義する**

書式 ON STRIG GOSUB [<行番号>] [, <行番号> …]

文例 ON STRIG GOSUB 1000, 2000

解説 スペースキーやジョイスティックのトリガが押されたときに実行されるサブルーチンの開始行を、<行番号>で定義します。行番号は最大5個まで並べることが可能で、左からスペースキー、ポート1に接続されたジョイスティックのトリガ1、ポート2に接続されたジョイスティックのトリガ1、ポート1に接続されたジョイスティックのトリガ2、ポート2に接続されたジョイスティックのトリガ2に対応しています。

指定したサブルーチンは、STRIG ON(割り込み許可)状態でスペースキーやジョイスティックのトリガが押されたときに、通常処理を中断する割り込み処理として実行されます。サブルーチンの実行中はSTRIG STOP(割り込み保留)状態になり、RETURN文によって割り込みを終了して通常処理に復帰するとSTRIG ON状態になります(二重割り込みの防止)。割り込みを禁止する場合は、STRIG OFFを実行してください。

参照 STRIG, STRIG ON, STRIG OFF, STRIG STOP, ON GOSUB

OPEN(オープン)ステートメント

機能 **ファイルを開く**

書式 OPEN <ファイルスペック> [FOR <モード>] AS # <ファイル番号> [LEN=<レコード長>]

文例 OPEN "TEMP.DAT" FOR INPUT AS #1

解説 <ファイルスペック>で指定するファイルを開き、<ファイル番号>に割り当てます。以後、<ファイル番号>を指定して、INPUT#やPRINT#で入出力を行えます。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。Disk BASICでデバイス名を省略すると、カレントドライブのファイルを開きます。Disk BASICが存在しない場合、デバイス名を省略することはできません。デバイス名に指定できる名前は、表2-2の通りです。

コンソールデバイス(CON)は、キーボードからの入力と画面への出力を行います。ヌルデバイス(NUL)は何も行わないデバイスで、入力時にはEOFが常に真を返し、出力したデータは捨てられます。コンソールデバイスとヌルデバイスは、OPEN以外でも使用できます(COPY[1]など)。

<モード>には、ファイルのアクセスモードを指定します。指定できるアクセスモードは、表2-3の通りです。

<ファイル番号>には、1～15の整数を使用できますが、MAX FILESで指定した値より大きな番号を指定することはできません。また、既に開かれているファイルのファイル番号を指定することはできません。ファイルを開くことに成功すると、以後、<ファイル番号>を用いてPRINT#やINPUT#などで入出力できます。

<レコード長>には、ランダムアクセスファイルとしてファイルを開いたときの最大レコード長を指定します。指定できる値の範囲は1～256で、省略すると256が指定されたものとみなされます。

OPEN文で開いたファイルを、再度開くことはできません。CLOSE文でファイルを閉じてから、再度開き直してください。

参照 CLOSE, PRINT#, INPUT#, INPUT$, EOF, MAX FILES

● **サンプルプログラム**

```
10 MAX FILES=1
20 COLOR 15,1,1:SCREEN 2
30 OPEN "GRP:" AS #1
40 DRAW "BM50,50"
50 PRINT #1,"Hello, world."
60 I$=INPUT$(1)
70 CLOSE #1
```

PAINT(ペイント)ステートメント

機能 **指定された境界色で囲まれた領域を塗りつぶす**

デバイス名	入出力デバイス	ファイル名	<モード>指定				備考
			INPUT	OUTPUT	APPEND	random	
CAS:	カセットテープ	6文字まで有効	○	○	×	×	
CRT:	テキスト画面	無効(無視される)	×	○	×	×	
GRP:	グラフィック画面	〃	×	○	×	×	
LPT:	プリンタ	〃	×	○	×	×	
MEM:	メモリディスク	8文字+拡張子3文字まで有効	○	○	×	×	MSX2以降のみ
A:～H:	ディスクデバイス	〃	○	○	○	○	Disk BASICでのみ有効
CON	コンソールデバイス	無効(指定不可)	○	○	○	○	〃
NUL	ヌルデバイス	〃	○	○	○	○	〃

表2-2 OPENステートメントでデバイス名に指定できる名前

<モード>	意味	使用可能なデバイス名	備考
INPUT	既存のシーケンシャルファイルに対して入力を行う	CAS:, MEM:, A:～H:, CON, NUL	ファイルが存在しない場合はエラー(File not found)になる
OUTPUT	シーケンシャルファイルを新規に作成して出力を行う	すべてのデバイス	既存のファイルを指定した場合は上書きする
APPEND	既存のシーケンシャルファイルに対して追記出力を行う	A:～H:, CON, NUL	ファイルが存在しない場合はエラー(File not found)になる
FOR <モード>を省略	ランダムアクセスファイルに対して入出力を行う	A:～H:, CON, NUL	ファイルが存在しない場合は新規に作成する

表2-3 OPENステートメントで指定できるファイルのアクセスモード

書式 PAINT [STEP] (x, y) [, <領域色>] [, <境界色>]

文例 PAINT (150,100),7

解説 座標(x, y)を含む<境界色>で囲まれた領域を、<領域色>で塗りつぶします。グラフィックモード(SCREEN 2～12)でのみ使用できますが、SCREEN 2と4では<境界色>が無視され、<領域色>と同じ色で囲まれた領域が塗りつぶされます。

(x, y)を最終参照点(LP)からの相対座標で指定する場合は、STEPを(x, y)の前に付加します。LPはPAINT文の実行後も変化しません。

<領域色>、<境界色>はカラーコードで指定します。<領域色>を省略するとCOLOR文で指定した前景色が、<境界色>を省略すると<領域色>と同じカラーコードが用いられます。

(x, y)で指定する点の色が<境界色>と同じ場合、PAINT文は塗りつぶしを行いません。また、(x, y)の周囲に<境界色>で囲まれた領域がない場合、画面全体を<領域色>で塗りつぶします。

参照 COLOR, SCREEN, LINE, CIRCLE, PSET, PRESET

● サンプルプログラム

```
10 SCREEN 5
20 CIRCLE (128,106),50,15
30 PAINT (128,106),4,15
40 CIRCLE STEP (0,0),10,9
50 PAINT STEP (0,0),6,9
60 I$=INPUT$(1)
```

PEEK(ピーク)関数

機能 メモリ上の指定された番地の値を得る

書式 PEEK(<数式>)

文例 A=PEEK(0)

解説 <数式>で指定する番地の値をメモリから読み出し、その結果を得ます。得られる値は0～255の整数です。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定でき、その値の範囲は-32768～32767(または0～65535)となります。

参照 POKE, VPEEK, VPOKE

PLAY *1 (プレイ)ステートメント

機能 音楽を演奏する

書式 PLAY <文字式1> [, <文字式2>] [, <文字式3>]

文例 PLAY "O4CDEFGABO5C"

解説 <文字式1>～<文字式3>に指定するミュージックマクロ言語(MML)に従って、音楽を演奏します。PLAY文では3つのチャンネルで同時に使用できるため、最大3重和音の演奏が可能です。チャンネル1～3は、それぞれ<文字式1>～<文字式3>に対応しています。

MMLには大別して、音程、音長、テンポ、音量、音色、ローカルマクロの6種類の命令があり、<文字式1>～<文字式3>に0個以上の命令を記述します。次に各MML命令の詳細を示します。

● 音程

A～G —— 音程を音階名で表現します。C～Gがハ長調のド～ソに、A～Bがラ～シに対応しています。また、A～Gの直後に+か#を付けると半音上がり、-を付けると半音下がります。

O <数値> —— 音階の音域を設定します。<数値>の範囲は1～8で、初期値は4です。<数値>が1上がると、音域が1オクターブ上がります。音域を設定すると、次にそのチャンネルで音域を設定するまで有効です。

N <数値> —— 音程を0～96の<数値>で指定します。N1がO1C#に対応し、<数値>が1増えるごとに半音上がるため、N95はO8Bと同じ音程になります。また、N0は休符(R命令)と同じ働きをします。

● 音長

L <数値> —— 音長を(1/<数値>)音に設定します。<数値>の範囲は1～64で、初期値は4です。L1が全音符に相当し、L2が2分音符、以後、64分音符まで続きます。音長を設定すると、次にそのチャンネルで音長を設定するまで有効です。また、1音のみ音長を変える場合、音階名(A～G)の直後に<数値>を指定します(例：PLAY "L8C"はPLAY "C8"と同じ)。

.(ピリオド) —— 音階名(A～G)や音程(N)、休符(R)の直後に付けて、音長を3/2倍にします。2個続けると9/4倍、3個で27/8倍になります。

R [<数値>] —— <数値>で指定する長さの休符を表します。<数値>の指定方法はL命令と同じです。<数値>を省略すると、4が指定されたものとみなされます。R命令は、L命令で設定した音長に影響されません。

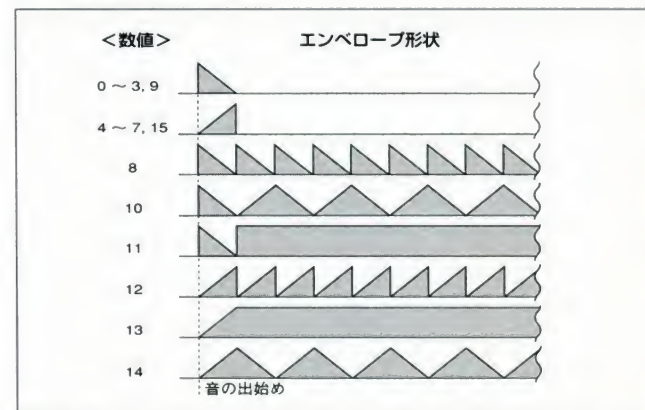
● テンポ

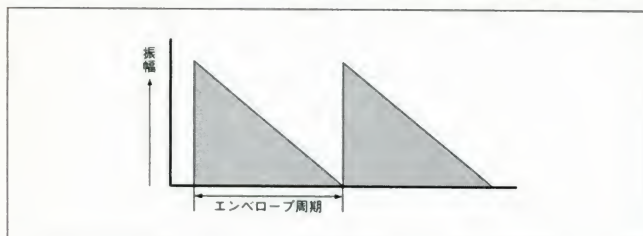
T <数値> —— 演奏のテンポを設定します。<数値>には、1分間に演奏する4分音符の数を指定します。<数値>の範囲は32～255で、初期値は120です。テンポを設定すると、次にそのチャンネルでテンポを設定するまで有効です。

● 音量

V <数値> —— 音量を設定します。<数値>の範囲は0～15で、初期値は8です。0のとき無音で、15のとき音量が最大になります。音量を設定すると、次のそのチャンネルで音量を設定するまで有効です。

● 音色

S <数値> —— エンベロープ形状(音量変化の波形)を設定します。1音ごとに、<数値>で指定するエンベロープ形状で音量が変化します。<数値>とエンベロープ形状の対応については、下図を参照してください。エンベロープ形状は、すべてのチャンネルで共通です。エンベロープ形状を設定すると、次回の設定まで有効です。また、エンベロープ形状の設定後に音量を設定すると、そのチャンネルのエンベロープ形状設定は無効になります。

M <数値> —— エンベロープ周期(音量変化の周期)を設定します。

<数値>の範囲は1～65535で、エンベロープ周期は(143 × <数値>) μ秒となります。エンベロープ周期を設定すると、次回の設定まで有効です。

● ローカルマクロ

X <文字型変数>; ―― 命令全体を<文字型変数>中のMML命令に置き換えます(例：A$="CDEFGAB": PLAY "O4XA$;O5C"はPLAY"O4 CDEFGABO5C"と同じ)。

= <数値型変数>; ―― 命令全体を<数値型変数>の値を表現する文字列に置き換えます(例：A=12:PLAY "V=A;CDEFGAB"はPLAY "V12 CDEFGAB"と同じ)。

参照 PLAY [*2]

PLAY [*2] (プレイ)関数

機能 音楽の演奏状態を調べる

書式 PLAY(<チャンネル番号>)

文例 A=PLAY(0)

解説 <チャンネル番号>で指定するチャンネルの演奏状態を調べ、演奏中であれば真(-1)を、そうでなければ偽(0)を得ます。<チャンネル番号>には、次の値を指定できます。

チャンネル番号	値
0	チャンネル1～3のいずれかが演奏中であれば-1を得る
1	チャンネル1が演奏中であれば-1を得る
2	チャンネル2が演奏中であれば-1を得る
3	チャンネル3が演奏中であれば-1を得る

参照 PLAY [*1]

PLAY#(プレイシャープ)ステートメント MUSIC

機能 MSX-MUSICを使って音楽を演奏する

書式 PLAY #<モード>, <文字式1> [, <文字式2> …]

文例 PLAY #2,"O4CDEFGABO5C"

解説 <文字式n>に指定するミュージックマクロ言語(MML)に従って、音楽を演奏します。<モード>に指定する値によって、同時に指定できる文字式の数が異なります(下表)。

<モード>	チャンネル数
0	MSX内蔵の音源チップProgramable Sound Generator(PSG)の3チャンネルに対応する<文字式1>～<文字式3>
1	使用不可(MSX-AUDIOとの互換性のために存在し、指定するとエラーになる)
2, 3	MSX-MUSIC(9チャンネルまたは6チャンネル+リズム音)とPSG(3チャンネル)に対応する<MSX-MUSIC文字式1>～<MSX-MUSIC文字式n>, <リズム音文字式>, <PSG文字式1>～<PSG文字式3>

MMLには大別して、音程、音長、テンポ、音量、音色、ローカルマクロ、リズム音の7種類の命令があり、各<文字式>に0個以上の命令を記述します。ただし、リズム音は<リズム音文字式>にのみ指定可能です。

<MSX-MUSIC文字式>は、CALL MUSICで指定した文字式の数だけ指定します。リズム音を使用しないモードでは、<リズム音文字式>をカンマと共に省略します。

<リズム音文字式>には、リズム音、テンポ、ローカルマクロ、R<数値>、@V <数値>命令を使用できます。

以下に、各MML命令の詳細を示します。

● 音程

A～G ―― 音程を音階名で表現します。C～Gがハ長調のド～ソに、A～Bがラ～シに対応しています。また、A～Gの直後に+か#を付けると半音上がり、-を付けると半音下がります。

O <数値> ―― 音階の音域を設定します。<数値>の範囲は1～8で、初期値は4です。<数値>が1上がると、音域が1オクターブ上がります。音域を設定すると、次にそのチャンネルで音域を設定するまで有効です。

> ―― 音階の音域を一つ上げます。PSG文字式でも使用できます。

< ―― 音階の音域を一つ下げます。PSG文字式でも使用できます。

N <数値> ―― 音程を0～96の<数値>で指定します。N1がO1C#に対応し、<数値>が1増えるごとに半音上がるため、N95はO8Bと同じ音程になります。また、N0は休符(R命令)と同じ働きをします。

● 音長

L <数値> ―― 音長を(1/<数値>)音に設定します。<数値>の範囲は1～64で、初期値は4です。L1が全音符に相当し、L2が2分音符、以後、64分音符まで続きます。音長を設定すると、次にそのチャンネルで音長を設定するまで有効です。また、1音のみ音長を変える場合、音階名(A～G)の直後に<数値>を指定します(例：PLAY "L8C"はPLAY "C8"と同じ)。

.(ピリオド) ―― 音階名(A～G)や音程(N)、休符(R)の直後に付けて、音長を3/2倍にします。2個続けると9/4倍、3個で27/8倍になります。

& ―― 二つの音(音階名(A～G)や音程(N)、休符(R))の間に置いて、前後の音をつないで演奏します(タイ)。二つの音は、同じ音階でなければなりません。PSG文字式でも使用できます。

{} <数値> ―― <数値>で指定される長さの音を、{}で囲まれた音(音階名(A～G)や音程(N))の数で等分して演奏します(連符)。<数値>の指定方法はL命令と同じです。<数値>を省略すると、L命令で設定した音長が指定されたものとみなされます。音の数と<数値>の積が64を越えるとエラーとなります。また、PSG文字式で指定するとエラーとなります。

R [<数値>] ―― <数値>で指定する長さの休符を表します。<数値>の指定方法はL命令と同じです。<数値>を省略すると、4が指定されたものとみなされます。R命令は、L命令で設定した音長に影響されません。

@W <数値> ―― <数値>で指定する長さだけ、発音状態を維持します。<数値>の指定方法はL命令と同じです。<数値>を省略すると、L命令で設定した音長が指定されたものとみなされます。PSG文字式で使用すると、R命令と同じ意味になります。

Q <数値> —— L命令などで設定した音長の内、実際に音が鳴っている時間の割合を(<数値>/8)に設定します。<数値>の範囲は1～8で、初期値は8です。PSG文字式で指定すると無視されます。

● テンポ

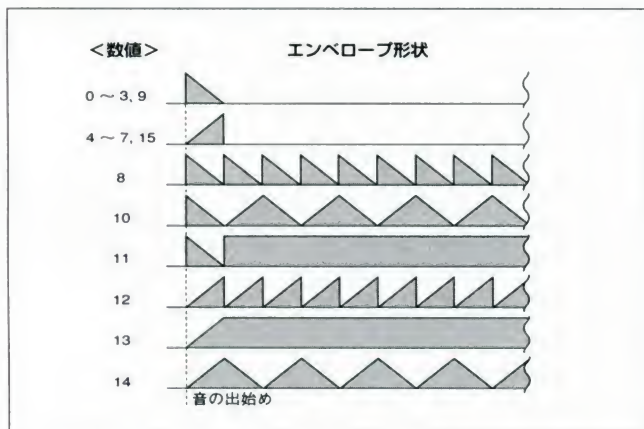
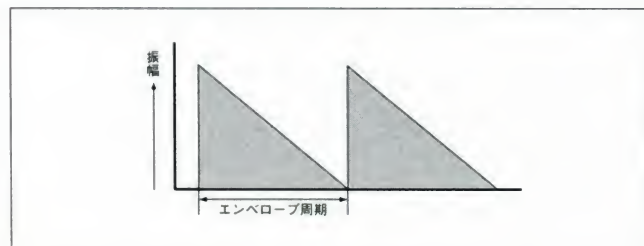
T <数値> —— 演奏のテンポを設定します。<数値>には、1分間に演奏する4分音符の数を指定します。<数値>の範囲は32～255で、初期値は120です。テンポを設定すると、次にそのチャンネルでテンポを設定するまで有効です。

● 音量

V <数値> —— 音量を設定します。<数値>の範囲は0～15で、初期値は8です。0のとき無音で、15のとき音量が最大になります。音量を設定すると、次のそのチャンネルで音量を設定するまで有効です。

@V <数値> —— 音量を細かく設定します。<数値>の範囲は0～127で、初期値は127です。V命令で指定した音量を、128段階に調整します。音量を設定すると、次のそのチャンネルで音量を設定するまで有効です。PSG文字式で指定すると無視されます。

● 音色

S <数値> —— エンベロープ形状(音量変化の波形)を設定します。1音ごとに、<数値>で指定するエンベロープ形状で音量が変化します。<数値>とエンベロープ形状の対応については、下図を参照してください。エンベロープ形状は、すべてのチャンネルで共通です。エンベロープ形状を設定すると、次回の設定まで有効です。また、エンベロープ形状の設定後に音量を設定すると、そのチャンネルのエンベロープ形状設定は無効になります。PSG文字式でのみ使用できます。

M <数値> —— エンベロープ周期(音量変化の周期)を設定します。<数値>の範囲は1～65535で、エンベロープ周期は(143 × <数値>)μ秒となります。エンベロープ周期を設定すると、次回の設定まで有効です。PSG文字式でのみ使用できます。

@ <数値> —— MSX-MUSICの音色を、<数値>によって指定します。<数値>の範囲は0～63です。音色と<数値>の対応については、CALL VOICEを参照してください。PSG文字式で指定すると無視されます。

Y <番号>,<数値> —— <番号>で指定される音源チップのレジスタに、<数値>を書き込みます。詳細は省略します。PSG文字式で指定すると無視されます。

● リズム音

B, S, M, C, H —— それぞれ、バスドラム音、スネアドラム音、タムタム音、シンバル音、ハイハット音を発声します。並べて書くことで、最大3音を同時発声することができます(例："BSH8MC4"はバスドラム音、スネアドラム音、ハイハット音を8分音符分、発声し、次にタムタム音、シンバル音を4分音符分、発声する)。

<数値> —— 直前の楽音を発声し、<数値>で指定する長さだけ待ちます。<数値>の指定方法はL命令と同じです。

! —— 直前のリズム音(B, S, M, C, H)の音量をアクセントボリュームにします。

V <数値> —— アクセントのない楽音の音量を設定します。<数値>の範囲は0～15で、初期値は8です。0のとき無音で、15のとき音量が最大になります。一度音量を設定すると、再設定するまで有効です。

@A <数値> —— アクセントのある楽音の音量を設定します。<数値>の範囲は0～15です。

● ローカルマクロ

X <文字型変数>; —— 命令全体を<文字型変数>中のMML命令に置き換えます(例：A$="CDEFGAB": PLAY #2,"O4XA$;"はPLAY #2,"O4CDEFGAB"と同じ)。同一PLAY文の同一文字式において、この命令以降に命令を書くことはできません。

= <数値型変数>; —— 命令全体を<数値型変数>の値を表現する文字列に置き換えます(例：A=12:PLAY #2,"V=A;CDEFGAB"はPLAY #2,"V12CDEFGAB"と同じ)。

参照 CALL MUSIC, CALL BGM, CALL PITCH, CALL PLAY, CALL STOPM, CALL TEMPER, CALL TRANSPOSE, CALL VOICE, CALL VOICE COPY

POINT(ポイント)関数

機能 指定した座標の色を得る

書式 POINT(x, y)

文例 A=POINT(100,100)

解説 (x, y)で指定する座標に表示されているドットの色を調べ、カラーコードで得ます。グラフィックモード(SCREEN 2～12)でのみ使用できます。

なお、(x, y)にスプライトが表示されている場合、スプライトの色ではなく、背景の色を得ます。

参照 COLOR

POKE(ポーク)ステートメント

機能 メモリ上の指定された番地にデータを書き込む

書式 POKE <番地>,<データ>

文例 POKE &HF3DB,0

解説 <番地>で指定するメモリ上の1バイトに<データ>の値を書き

込みます。<番地>には、整数型、単精度実数型、倍精度実数型の値を指定でき、その値の範囲は-32768～32767(または0～65535)となります。<データ>に指定できる値の範囲は0～255の整数です。

参照 PEEK, VPOKE, VPEEK

POS(ポジション)関数

機能 現在のカーソル桁位置を得る

書式 POS(<数式>)

文例 X=POS(0)

解説 関数を実行した時点のカーソルの桁位置を得ます。画面左端の桁を0とし、画面右へ向かうに従って大きくなる値を返します。<数式>の値は無視されます(意味を持ちません)。

テキストモード(SCREEN 0～1)でのみ、意味のある値が得られます。

参照 CSRLIN, LOCATE

PRESET(ポイント リセット)ステートメント

機能 指定する座標の点を消去する

書式 PRESET [STEP] (x, y), [, <カラーコード>] [, <論理演算子>]

文例 PRESET (100,100)

解説 (x, y)で指定する座標に描かれた点を消去します。グラフィックモード(SCREEN 2～12)でのみ使用できます。

(x, y)を最終参照点(LP)からの相対座標で指定する場合は、STEPを(x, y)の前に付加します。LPはPRESETの実行後、(x, y)に設定されます。

消去する際に使用する色を<カラーコード>で指定します。<カラーコード>を省略するとCOLOR文で指定した背景色が用いられます。

<論理演算子>を指定すると、<カラーコード>を転送元のカラーコードとする論理演算を行います。論理演算子は、SCREEN 5～12でのみ使用できます。論理演算子の詳細は、COPY [*2]を参照してください。

参照 PSET, COPY [*2], COLOR

PRINT(プリント)ステートメント

機能 数値や文字列を画面に表示する

書式 PRINT [<式> …]

文例 PRINT "A$=";A$,"B%=";B%

解説 <式>で指定する数値や文字列を画面に表示します。テキストモード(SCREEN 0～1)でのみ、有効です。<式>を省略すると、改行のみを行います。

複数の<式>を指定する場合、<式>をカンマ(,)かセミコロン(;)で区切ります。カンマで区切ると、14桁ごとに表示されます。セミコロンで区切ると、直前に表示したもののすぐ後に表示されます。最後の<式>の後ろにセミコロンがあると、次のPRINT文は改行せずに続けて表示します。

数値を表示する場合、その後ろに空白が1文字分表示されます。また、符号を表示するための桁が数値の前に確保され、正の場合は空白を、負の場合はマイナス符号(-)を表示します。数値が単精度型、倍精度型の場合、それぞれ6桁、14桁までは実数形式で表示しますが、それ以上の桁数を持つ値は指数形式で表示します。

PRINTの省略形として疑問符(?)を使用できます。

参照 PRINT USING, LOCATE

● サンプルプログラム

```
10 A=10:B=35:A$="hello":B$="world"
20 PRINT A,B
30 PRINT A;B
40 PRINT "A+B"
50 PRINT A+B
60 PRINT A$,B$
70 PRINT A$;B$
80 PRINT "A$+B$"
90 PRINT A$+B$
100 PRINT
110 PRINT A;
120 PRINT B
130 PRINT A$;
140 PRINT B$
```

PRINT USING(プリント ユージング)ステートメント

機能 数値や文字列を書式化して画面に表示する

書式 PRINT USING <書式化文字列>; <式> …

文例 PRINT USING "###.#### & &";32.16597,"ABCDEF",10

実行結果 32.1660 ABC 10.0000

解説 <式>で指定する数値や文字列を、<書式化文字列>で指定する書式に従って編集し、画面に表示します。テキストモード(SCREEN 0～1)でのみ、有効です。

複数の<式>を指定する場合、<式>をカンマ(,)かセミコロン(;)で区切ります。カンマで区切ってもコロンで区切っても同じ結果になりますが、最後の<式>の後ろにセミコロンがあると、次のPRINT文、PRINT USING文は改行せずに続けて表示します。

<書式化文字列>の中に複数の書式を指定すると、先頭の書式によって最初の<式>が編集され、以後、順に2番目以降の書式と<式>を組み合わせて編集されます。<式>の数が書式よりも多い場合、すべての書式を使い切った次の<式>から再度、先頭の書式を使います。

<書式化文字列>に指定する書式は、以下の書式制御文字を組み合わせた文字列です。文字列を編集するものと、数値を編集するものがあります。グラフィックキャラクタは、1文字当たり2文字として数えられます。

● 文字列を編集する書式制御文字

! ── <式>で指定する文字列の先頭1文字を表示する。

例：PRINT USING "!";"ABC"

実行結果　A

&<n個の空白>& ── <式>で指定する文字列の先頭(n + 2)文字を表示する。文字列が(n + 2)より長い場合は切り捨てられ、短い場合は空白が末尾に補われる。

例：PRINT USING "& &";"A","ABCDEF"

実行結果　A ABC

@ ── <式>で指定する文字列を表示する。<書式化文字列>中の@の数が<式>の数と合わずに余った場合、余分の@と@以降の文字は無視される。

例：PRINT USING "@,@!";"hello","world"," test"

実行結果　hello,world! test,

● 数値を編集する書式制御文字

── <式>で指定する数値を、#の数で指定する桁数だけ表示する。数値の桁数が指定よりも小さければ右詰で表示され、左側に空白が補われる。また、書式に指定した桁数より数値の桁数の方が大きい場合、数値の直前に%が表示される。

例：PRINT USING " value=#####";1234,123456

実行結果　value= 1234 value=%123456

. ── 小数点の位置を指定する。<式>で指定する数値の小数部の桁数が指定よりも小さければ、右側に0が補われる。

例：PRINT USING " value=#.###";5.96,9.9999

実行結果　value=5.960 value=%10.000

+ ── 符号桁を確保し、正負符号を表示する。書式の左端に指定すると数値の前に、右端に指定すると後ろに符号桁が表示される。両端に指定した場合、右端に指定したものは制御文字とはみなされない。

例：PRINT USING "+# #+ ";1,-1,-1,1

実行結果　+1 1- -1 1+

– ── 書式の右端に指定すると、数値の後ろに符号桁を確保して空白(<式>が正の場合)か負符号(<式>が負の場合)を表示する。書式の左端に指定したものは、制御文字とはみなされない。

例：PRINT USING "#- ";1,-1

実行結果　1 1-

＊＊ ── 書式左端の"##"の代わりに指定すると、整数部の桁数が指定よりも小さい場合に補われる文字をアスタリスク(＊)にする。整数部の書式が3桁以上の場合に使用できる。

例：PRINT USING "＊＊##.###";123.45

実行結果　＊123.450

¥¥ ── 書式左端の"##"の代わりに指定すると、数値の直前に¥を表示する。"¥¥"は"##"と同じく2桁を確保するが、その内の1桁を¥の表示に使用する。整数部の書式が2桁以上の場合に使用できる。

例：PRINT USING "¥¥##.###";123.45,67.8

実行結果　¥123.450 ¥67.800

＊＊¥ ── 書式左端の"###"の代わりに指定すると、整数部の桁数が指定よりも小さい場合に補われる文字をアスタリスク(＊)にし、数値の直前に¥を表示する。"＊＊¥"は"###"と同じく3桁を確保するが、その内の1桁を¥の表示に使用する。整数部の書式が3桁以上の場合に使用できる。

例：PRINT USING "＊＊¥##.###";123.45,67.8

実行結果　¥123.450＊¥67.800

, ── 書式の整数部右端に指定すると、数値の整数部を3桁ごとにカンマで区切って表示する。小数点より右側に指定した場合や、指数形式の書式("^^^^")を指定している場合は無効になる。

例：PRINT USING "#####, ####,.###";12345,6789.01

実行結果　12,345 6,789.010

^^^^ ── 書式の右側に指定すると、数値を指数形式で表示する。

例：PRINT USING "+#.####^^^^ ＊＊##.###^^^^";12345,6789.01

実行結果　+0.1235E+05 ＊678.901E+01

参照 PRINT, LOCATE

PRINT#(プリント シャープ)ステートメント

機能 数値や文字列をファイルに出力する

書式 PRINT #<ファイル番号>, [<式> …]

文例 PRINT #1,"A$=";A$,"B%=";B%

解説 <ファイル番号>で指定するファイルへ、<式>で指定する数値や文字列を出力します。ファイルは、OPEN文のOUTPUTモードかAPPENDモードで開かれている必要があります。

PRINT#文の出力形式は、PRINT文が画面へ出力するときの形式と同じです。

参照 PRINT, PRINT# USING, INPUT#, LINE INPUT#, INPUT$, OPEN, CLOSE

PRINT# USING(プリント シャープ ユージング)ステートメント

機能 数値や文字列を書式化してファイルに出力する

書式 PRINT #<ファイル番号>, USING <書式化文字列>; <式> …

文例 PRINT #1,USING "###.#### & &";32.16597,"ABCDEF",10

解説 <式>で指定する数値や文字列を、<書式化文字列>で指定する書式に従って編集し、<ファイル番号>で指定するファイルへ出力します。ファイルは、OPEN文のOUTPUTモードかAPPENDモードで開かれている必要があります。

PRINT# USING文の出力形式は、PRINT USING文が画面へ出力するときの形式と同じです。

参照 PRINT USING, PRINT#, INPUT#, LINE INPUT#, INPUT$, OPEN, CLOSE

PSET(ポイント セット)ステートメント

機能 指定する座標に点を描く

書式 PSET [STEP] (x; y), [, <カラーコード>] [, <論理演算子>]

文例 PSET (100,100)

解説 (x, y)で指定する座標に点を描きます。グラフィックモード(SCREEN 2～12)でのみ使用できます。

(x, y)を最終参照点(LP)からの相対座標で指定する場合は、STEPを(x, y)の前に付加します。LPはPSETの実行後、(x, y)に設定されます。

点を描く際に使用する色を<カラーコード>で指定します。<カラーコード>を省略するとCOLOR文で指定した前景色が用いられます。

<論理演算子>を指定すると、<カラーコード>を転送元のカラーコードとする論理演算を行います。論理演算子は、SCREEN 5～12でのみ使用できます。論理演算子の詳細は、COPY [*2]を参照してください。

参照 PSET, COPY [*2], COLOR

PUT(プット)ステートメント　Disk

機能 ランダムアクセスファイルに入出力バッファ中のレコードを書き込む

書式 PUT [#]<ファイル番号> [, <レコード番号>]

文例 PUT #1,RN

解説 <ファイル番号>で指定するランダムアクセスファイルに、入出力バッファ中のレコードを書き込みます。<レコード番号>を指定するとそのレコードに、省略すると最後に実行したPUT文、GET文で使用したレコードの次のレコードに書き込みを行います。

書き込みを行うレコードの内容は、FIELD文、LSET文、RSET文によって入出力バッファ中に構成しておかなければなりません。

参照 OPEN, FIELD, LSET, RSET, GET

PUT SPRITE(プットスプライト)ステートメント

機能 スプライトパターンを画面に表示する

書式 PUT SPRITE <スプライト面番号> [, [STEP] (x, y)] [, <カラーコード>] [, <スプライトパターン番号>]

文例 PUT SPRITE 1,(150,100),15,0

解説 <スプライト面番号>のスプライト面に、<スプライトパターン番号>で指定するスプライトパターンを表示します。

<スプライト面番号>には0～31の整数を指定します。各スプライト面には、1つのスプライトパターンのみを表示できます。そのため、PUT SPRITEを実行すると、同一スプライト面に表示されていたスプライトは消去され、新たなスプライトパターンのみが表示されます。

すべてのスプライト面に同時にスプライトパターンを表示できますが、水平方向に並べられるスプライトパターンは4個までで、5個目以降は表示されません(SCREEN 4以上では8個まで)。また、スプライト面には優先順位があり、スプライトパターンが重なった場合、スプライト面番号の小さなスプライト面のスプライトパターンが手前に表示されます。

(x, y)には、スプライトパターンを表示する座標を指定します。スプライトパターンの左上角の座標が(x, y)となるように表示されます。xには-32～255の整数を、yには-32～191(SCREEN 4以上では-32～211)の整数を指定します。SCREEN 6～7では水平方向の解像度が増しますが、スプライトの座標系やスプライトパターンの横幅は変わりません。(x, y)を省略すると、指定したスプライト面の座標がそのまま用いられます。

yに208(SCREEN 4以上では216)を指定すると、そのスプライト面以降のスプライトパターンをすべて消去します。yに209(SCREEN 4以上では217)を指定すると、そのスプライト面のスプライトパターンのみを消去します。

(x, y)を最終参照点(LP)からの相対座標で指定する場合は、STEPを(x, y)の前に付加します。SCREEN 6～7で相対座標を指定する場合、LPのX座標にxを足して256で割った余りがスプライトパターンの表示X座標になります。LPはPUT SPRITEの実行後、(x, y)に設定されます。

<カラーコード>には、スプライトパターンの表示色を指定します。<カラーコード>を省略するとCOLOR文で指定した前景色が用いられますが、SCREEN 4以上でCOLOR SPRITEやCOLOR SPRITE$を使用した場合は、その設定色が用いられます(COLOR SPRITEやCOLOR SPRITE$の設定色を有効にするには、<カラーコード>を省略する必要があります)。また、SCREEN 8のカラーコードについては、COLOR SPRITE$を参照してください。

<スプライトパターン番号>は、SPRITE$で定義したスプライトパターンの番号です。<スプライトパターン番号>を省略すると、指定したスプライト面に表示中のスプライトパターンが用いられます。

スプライトパターンの大きさは、8×8ドット、16×16ドット(8×8ドットの倍角)、16×16ドット、32×32ドット(16×16ドットの倍角)から選べ、SCREEN文の第2パラメータで設定します。

参照 SPRITE$, COLOR SPRITE, COLOR SPRITE$

READ(リード)ステートメント

機能 DATA文で定義したデータを読み出し、変数に代入する

書式 READ <変数名> [, <変数名> …]

文例 READ A,B,C$

解説 データ文で定義したデータを読み込み、<変数名>で指定する変数に代入します。READ文は、行番号の小さいDATA文中のデータから順に読み込み、<変数名>の変数に代入していきます。

<変数名>の並びの数がDATA文で定義したデータの数より多い場合、エラー(Out of DATA)となります。また、<変数名>の並びの数がDATA文中のデータの数より少ない多い場合、残ったデータは次のREAD文で読み込まれます。

<変数名>で指定する変数の型に制限はありませんが、対応するDATA文のデータの型と一致するか、変換可能である必要があります。

参照 DATA, RESTORE

REM(リマーク)ステートメント

機能 注釈を入れる

書式 REM [<注釈文>]

文例 REM comment

解説 プログラム中に<注釈文>で指定する注釈を入れます。REM文はプログラムの実行に影響を与えませんが、GOTOやGOSUBの飛び先として使用できます。

REMの省略形としてアポストロフィー(')を使用できます。

RENUM(リナンバー)ステートメント

機能 行番号を振りなおす

書式 RENUM [<新行番号>] [, <旧行番号>] [, <増分>]

文例 RENUM 1000,630

解説 プログラムの行番号を振りなおします。

<新行番号>は、新たに指定する最初の行番号です。省略すると、10が指定されたものとして扱われます。

<旧行番号>は、番号の振りなおしを始める行番号です。省略すると、プログラムの先頭から降りなおします。

<増分>は、新たな行番号の間隔です。省略すると、10が指定されたものとして扱われます。

RENUM文は、GOTOやGOSUBなどで参照している行番号の更新も行います。もし、参照している行番号の行が存在しない場合、エラー(Undefined line XXXXX in YYYYY)となり、エラーとなった行番号は更新されません。

参照 LIST, DELETE, NEW, MERGE

RESTORE(リストア)ステートメント

機能 READ文で読み出すDATA文を指定する

書式 RESTORE [<行番号>]

文例 RESTORE 1000

解説 READ文が読み出しを行うDATA文を<行番号>で指定します。<行番号>を省略すると、最初のDATAが指定されたものとして扱われます。

参照 READ, DATA

RESUME(リシューム)ステートメント

機能 エラー処理ルーチンから復帰する

書式 RESUME [<行番号>]
RESUME NEXT

文例 RESUME 0

解説 エラー処理ルーチンを終了し、通常処理に復帰します。

<行番号>に0を指定するとエラーの発生した文から、0以外の値を指定するとその行から実行を再開します。<行番号>を省略すると、0が指定されたものとみなされます。

RESUME NEXTを実行すると、エラーの発生した文のすぐ次の文から実行を再開します。

参照 ON ERROR GOTO

RETURN(リターン)ステートメント

機能 サブルーチンから呼び出し元に制御を戻す

書式 RETURN [<行番号>]

文例 RETURN

解説 サブルーチンを終了し、呼び出し元の次の文に制御を戻します。<行番号>を指定すると、サブルーチンを終了した後、指定した行へ強制的に制御を移します。

参照 GOSUB, ON GOSUB

RIGHT$(ライトダラー)関数

機能 文字列の右側から、指定された長さの文字列を得る

書式 RIGHT$(<文字式>, <数式>)

文例 PRINT RIGHT$("ABCDEFG",3)
実行結果　EFG

解説 <文字式>で表される文字列の右側(末尾)から、<数式>の値の文字数分の文字列を得ます。得られる値は文字型です。<数式>に指定できる値の範囲値は、0～255です。

<文字式>の総文字数より<数式>の値が大きい場合、<文字式>全体を得ます。また、<数式>の値が0の場合、常にヌルストリング("")を得ます。

グラフィックキャラクタは、1文字当たり2文字として数えられます。

参照 LEFT$, MID$ *1

RND(ランダム)関数

機能 乱数を得る

書式 RND(<数式>)

文例 R=RND(-1)

解説 0以上1未満の乱数を発生し、その値を得ます。

<数式>が正の場合、同じ乱数系列の新しい乱数を得ます。この場合、符号が正であるということのみが意味を持ち、<数式>の値は無視されます。

<数式>が0の場合、直前の乱数を得ます。

<数式>が負の場合、<数式>の値に対応する乱数系列を作成し、その値を得ます。

● サンプルプログラム

```
10 SCREEN 0:WIDTH 40:KEY OFF
20 R=RND(-TIME)
30 FOR I=0 TO 255
40 X=RND(1)*40
50 Y=RND(1)*22
60 CH$=CHR$(RND(1)*224+32)
70 LOCATE X,Y:PRINT CH$;
80 NEXT
90 I$=INPUT$(1)
```

RSET(ライトセット)ステートメント　Disk

機能 ランダムアクセスファイルの入出力バッファに、データを右詰で転送する

書式 RSET <文字型変数名>=<文字式>

文例 RSET NM$="MSX"

解説 <文字型変数名>で指定するランダムアクセスファイルのフィールドに、<文字式>で表される文字列を転送します。転送はランダムアクセスファイルの入出力バッファに対して行われ、実際にファイルに出力されるのはPUT文の実行時です。

<文字式>の文字列の長さがFIELD文で指定したフィールド幅よりも小さい場合、右詰で転送され、余った部分は空白で埋められます。文字列の長さがフィールド幅よりも大きい場合は、文字列の右側が切り捨てられます。

RSETは文字型のデータしか扱えないため、数値型のデータをフィールドに使用する場合はMKI$、MKS$、MKD$を使って文字型に変換する必要があります。

参照 OPEN, FIELD, PUT, LSET, MKI$, MKS$, MKD$

RUN *1 (ラン)ステートメント

機能 プログラムを実行する

書式 RUN [<行番号>]

文例 RUN

解説 プログラムを<行番号>で指定する行から実行します。<行番号>を省略すると、プログラムの先頭から実行を始めます。

RUN *2 (ラン)ステートメント　Disk

機能 ファイルに保存されたプログラムをメモリ上に読み込んで実行する

書式 RUN <ファイルスペック> [, R]

文例 RUN "SAMPLE.BAS"

解説 <ファイルスペック>で指定するファイルを、BASICプログラムとしてメモリ上に読み込み、実行します。RUN文を実行すると、RUN文実行前にメモリ上にあったプログラムは消去されます。また、開かれているすべてのファイルが閉じられ、変数の値が初期化されます。ただし、Rオプションを指定すると、既に開かれているファイルがあっても閉じません。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブのファイルを読み出します。

RUN文は指定されたファイルを見つけてプログラムのロードを始めるまでは、メモリ上のプログラムを消去しません。

参照 RUN *1, LOAD, SAVE

SAVE(セーブ)ステートメント

機能 メモリ上のプログラムをファイルに保存する

書式 SAVE <ファイルスペック> [, A]

文例 SAVE "SAMPLE.BAS"

解説 メモリ上のBASICプログラムを、<ファイルスペック>で指定するファイルに保存します。Disk BASICでAオプションを指定すると、アスキー形式で保存されます。Disk BASICが存在しない場合は、Aオプションを指定することはできず、常にアスキー形式で保存されます。アスキー形式で保存すると、MERGE文でプログラムを混合することができます。

<ファイルスペック>は"デバイス名:ファイル名"の形式で指定します。デバイス名を省略すると、カレントドライブのファイルに保存します(Disk BASICが存在する場合)。

参照 LOAD, MERGE

SCREEN(スクリーン)ステートメント

機能 スクリーンモードやスプライトサイズなどの設定を行う

書式 SCREEN [<スクリーンモード>] [, <スプライトサイズ>] [, <キークリック>] [, <カセットボーレート>] [, <プリンタ種別>] [, <インターレースモード>]

文例 SCREEN 5,2,,,,0

解説 画面表示に関係するモードや、キークリック音の有無、カセットテープへの書き込み速度、プリンタ種別を設定します。

<スクリーンモード>には、0～12の値を持つ数式を指定します。ただし、MSX1では0～3、VRAMが64KBのMSX2では0～6、VRAMが128KBのMSX2では0～8のみが指定できます。各モードの詳細は、表2-4の通りです。起動直後には、1が設定されています。

<スプライトサイズ>には、0～3の値を持つ数式を指定します。設定値の詳細は下表の通りです。<スプライトサイズ>を指定すると、定義済みのスプライトパターンやスプライトの座標などが初期化されます。起動直後には、0が設定されています。

スプライトサイズ	詳細
0	8×8ドットのスプライトを表示する
1	8×8ドットのスプライトを縦横2倍に拡大して表示する
2	16×16ドットのスプライトを表示する
3	16×16ドットのスプライトを縦横2倍に拡大して表示する

<キークリック>には、キークリック音の有無を、0～1の値を持つ数式で指定します。0を指定するとキークリック音を出さない設定に、1を指定するとキークリック音を出す設定にします。起動直後には、1が設定されています。

<カセットボーレート>には、カセットテープへ書き込むときの転送速度(ボーレート)を、1～2の値を持つ数式で指定します。1を指定すると1200ボーに、2を指定すると2400ボーに設定します。この設定値は、CSAVEでボーレートを省略した場合に用いられます。起動直後には、1が設定されています。

<プリンタ種別>には、使用しているプリンタがMSX仕様であるかどうかを、0～255の値を持つ数式で指定します。0を指定するとMSX仕様のプリンタに、0以外を指定するとMSX仕様でないプリンタに設定します。MSX仕様でないプリンタの場合、グラフィック文字は空白に、半角平仮名は半角片仮名に変換して出力します。起動直後には、0が設定されています。

<インターレースモード>には、画面のインターレースモードを、0～3の値を持つ数式で指定します。設定値の詳細は表2-5の通りです。このパラメータはMSX2以降でのみ使用できます。起動直後には、0が設定されています。なお、ディスプレイページについては、SET

スクリーンモード	解像度	色数	備考
0	最大80桁×24桁のテキストモード	前景色、背景色に各1色	MSX1では最大40桁、文字を横6ドットで表示する
1	最大32桁×24桁のテキストモード	前景色、背景色、周辺色に各1色	スプライトの使用可
2	256×192ドットのグラフィックモード	横8ドットごとに16色中の2色	MSX2以降ではパレットの使用可
3	64×48ブロックのグラフィックモード	1ブロック(4×4ドット)ごとに16色中の1色	MSX2以降ではパレットの使用可
4	256×192ドットのグラフィックモード	横8ドットごとに16色中の2色	COLOR SPRITE, COLOR SPRITE$の使用可
5	256×212ドットのグラフィックモード	1ドットごとに16色中の1色	パレットの使用可
6	512×212ドットのグラフィックモード	1ドットごとに4色中の1色	パレットの使用可
7	512×212ドットのグラフィックモード	1ドットごとに16色中の1色	パレットの使用可
8	256×212ドットのグラフィックモード	1ドットごとに256色中の1色	固定256色
10	256×212ドットのグラフィックモード	YJKとRGBの混在でグラフィック処理あり	パレットの使用可
11	256×212ドットのグラフィックモード	YJKとRGBの混在でグラフィック処理なし	パレットの使用可
12	256×212ドットのグラフィックモード	YJK	

表2-4 スクリーンモード

インターレースモード	走査モード	表示ページ	ディスプレイページ	備考
0	ノンインターレース	ディスプレイページを表示	任意	通常のモード
1	インターレース	ディスプレイページを表示	任意	走査線の隙間を直前の走査線と同じデータで埋める
2	ノンインターレース	ディスプレイページとその直前のページを交互に表示	奇数ページ(1か3)	2画面を1/30秒ごとに切り替えて表示する
3	インターレース	ディスプレイページとその直前のページを交互に表示	奇数ページ(1か3)	擬似的に垂直方向の解像度を2倍にする

表2-5 インターレースモード

PAGEを参照してください。

参照 COLOR, COLOR SPRITE, COLOR SPRITE$, CSAVE, SET PAGE

SET ADJUST(セットアジャスト)ステートメント MSX2

機能 画面の表示位置を設定する

書式 SET ADJUST (x, y)

文例 SET ADJUST (-1, 3)

解説 画面の表示位置を(x, y)で指定する位置に移動します。x、yには、それぞれ-7～8の値を持つ数式を指定します。一度設定した値はシステムに記憶され、電源を切っても保持されます。

SET BEEP(セットビープ)ステートメント MSX2

機能 BEEPで鳴らす音の種類と音量を設定する

書式 SET BEEP [<音色>] [, <音量>]

文例 SET BEEP 1,4

解説 BEEP文で鳴らす音の種類と音量を設定します。<音色>、<音量>には、それぞれ1～4の値を持つ数式を指定します。<音量>は、値が大きいほど音も大きくなります。一度設定した値はシステムに記憶され、電源を切っても保持されます。

SET DATE(セットデート)ステートメント MSX2

機能 システムの日付を設定する

書式 SET DATE <日付> [, A]

文例 SET DATE "03/01/31"

解説 システムの日付を設定します。<日付>には、年月日を2桁ずつスラッシュ(/)で区切った形式の文字列を指定します(例：02/08/31)。Aオプションを指定すると、アラームの日付を設定します。

参照 SET TIME, GET DATE, GET TIME, TIME

SET PAGE(セットページ)ステートメント MSX2

機能 使用する画面のページを指定する

書式 SET PAGE [<ディスプレイページ>] [, <アクティブページ>]

文例 SET PAGE 1,3

解説 表示や読み書きに使用する画面のページを指定します。<ディスプレイページ>には表示するページを、<アクティブページ>には描画やデータの読み込みを行うページをページ番号で指定します。ページ番号に指定できる値は、次の表の通りです。

スクリーンモード	VRAM容量	
	64KB	128KB
0～4	使用不可	使用不可
5～6	0～1	0～3
7～12	---	0～1

参照 SCREEN

SET PASSWORD(セットパスワード)ステートメント MSX2

機能 システムにパスワードを設定する

書式 SET PASSWORD <パスワード>

文例 SET PASSWORD "Jh48IM23ODe"

解説 <パスワード>で指定する文字列(255文字以内)を、システムのパスワードとして設定します。パスワードを設定すると、パスワードを入力しなければシステムを起動時できないようになります。なお、[GRAPH]キーと[STOP]キーを押しながらシステムを起動すると、パスワードを入力せずに起動できます。また、SET PROMPTかSET TITLEを実行すると、パスワードは無効になります。

参照 SET PROMPT, SET TITLE

SET PROMPT(セットプロンプト)ステートメント MSX2

機能 BASICのプロンプトを設定する

書式 SET PROMPT <プロンプト>

文例 SET PROMPT "Ready"

解説 コマンドレベルで表示されるプロンプト(標準では"Ok")を、<プロンプト>で指定する文字列(6文字以内)に設定します。一度設定した値はシステムに記憶され、電源を切っても保持されます。なお、SET PASSWORDかSET TITLEを実行すると、プロンプトは"Ok"に戻ります。

参照 SET PASSWORD, SET TITLE

SET SCREEN(セットスクリーン)ステートメント MSX2

機能 表示パラメータを初期値として設定する

書式 SET SCREEN

文例 SET SCREEN

解説 SCREEN, WIDTH, COLOR, KEY ON [*1], KEY OFF [*1]で設定した値を、初期値として設定します(SCREENのスプライトサイズを除く)。一度設定した値はシステムに記憶され、次の起動時にその表示パラメータで起動します。

参照 SCREEN, WIDTH, COLOR, KEY ON [*1], KEY OFF [*1]

SET SCROLL(セットスクロール)ステートメント MSX2+

機能 ハードウェアスクロールを行う

書式 SET SCROLL [<X座標>] [, <Y座標>] [, <マスクモード>] [, <ページモード>]

文例 SET SCROLL 50,150,1,1

解説 画面全体をスクロールします。<X座標>には画面左への、<Y座標>には画面上へのスクロール量を数式で指定します。<X座標>、<Y座標>に指定する値の範囲は、それぞれ0～511、0～255の整数です。<X座標>、<Y座標>を省略すると、最後に指定したスクロール量が設定されます。<マスクモード>に1を指定すると、画面左の8ドットを表示しません。表示する場合は0を指定します。<マスクモード>を省略すると、0が指定されたものとして扱われます。

ディスプレイページを奇数ページに設定して<ページモード>に1を指定すると、連続する2ページ(2Nページと(2N+1)ページ)を横に連結してスクロールします。

0を指定した場合やディスプレイページが偶数ページの場合は、1ページのみのスクロールになります。<ページモード>は、SCREEN 5以上でのみ使用できます。

参照 SCREEN, SET PAGE

● サンプルプログラム

```
10 SCREEN 7
20 CIRCLE (150,100),50,14
30 SET PAGE 1,1
40 CLS
50 LINE (50,50)-(150,100),15,BF
60 FOR I=0 TO 511
70 SET SCROLL I,I MOD 256,1,1
80 NEXT
90 I$=INPUT$(1)
```

SET TIME(セットタイム)ステートメント　MSX2

機能 システムの時刻を設定する

書式 SET TIME <時刻> [, A]

文例 SET TIME "01:57:12"

解説 システムの時刻を設定します。<時刻>には、時分秒を2桁ずつコロン(:)で区切った形式の文字列を指定します(例：23:04:57)。Aオプションを指定すると、アラームの時刻を設定します。

参照 SET TIME, GET DATE, GET TIME, TIME

SET TITLE(セットタイトル)ステートメント　MSX2

機能 起動時に表示されるタイトルを設定する

書式 SET TITLE [<タイトル>] [, <背景色>]

文例 SET TITLE "MSX3"

解説 起動時に表示されるタイトルと背景色を設定します。

<タイトル>に指定する文字列は6文字以内ですが、6文字ちょうどの文字列を設定すると、何かキーを押すまでタイトル画面で止まるようになります。<タイトル>を省略すると、<背景色>のみを指定することができます。

<背景色>には、1～4の値を持つ数式を指定します。1は青、2は緑、3は赤、4はオレンジに対応します。<背景色>を省略すると、青(1)が指定されたものとして扱われます。

一度設定した値はシステムに記憶され、電源を切っても保持されます。なお、SET PASSWORDかSET PROMPTを実行すると、タイトルは無効になりますが、背景色は保持されます。

参照 SET PASSWORD, SET PROMPT

SGN(サイン)関数

機能 数式の符号を得る

書式 SGN(<数式>)

文例 PRINT SGN(3.14)
実行結果 1

解説 <数式>の符号を調べ、正なら1を、0なら0を、負なら-1を得ます。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できます。

SIN(サイン)関数

機能 正弦(サイン)を得る

書式 SIN(<数式>)

文例 PRINT SIN(3.14159266/4)＊2
実行結果 1.4142135646395

解説 <数式>正弦(サイン)を得ます。<数式>はラジアン単位で、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数です。

参照 COS, TAN, ATN

SOUND(サウンド)ステートメント

機能 音源チップのレジスタに値を設定する

書式 SOUND <レジスタ番号>, <データ>

文例 SOUND 0,254:SOUND 1,0:SOUND 8,8:SOUND 7,38:SOUND 12,13:SOUND 12,13

解説 MSX内蔵の音源チップProgramable Sound Generator(PSG)のレジスタを、直接書き換えます。書き換えるレジスタを<レジスタ番号>に、その内容を<データ>に指定します。<レジスタ番号>、<データ>に指定する値の範囲は、それぞれ0～13、0～255の整数です。

レジスタには、レジスタ番号ごとに機能が割り当てられています(下表)。

チャンネル番号	周波数	ノイズ周波数	チャンネル設定	音量	エンベロープ形状	エンベロープ周期
チャンネル1	0, 1	6	7	8	13	11, 12
チャンネル2	2, 3	6	7	9	13	11, 12
チャンネル3	4, 5	6	7	10	13	11, 12

レジスタに割り当てられた各機能は、以下の通りです。

● **周波数**

音の周波数をチャンネルごとに設定します。チャンネル1～3に、レジスタ0～1、2～3、4～5が割り当てられています。設定値の範囲は0～4095で、偶数番号のレジスタ、奇数番号のレジスタをそれぞれ下位8bit、上位4bitとして設定します。設定する周波数をFT [Hz]、設定値をRTとすると、RT=111860.78125/FTとなります。

● **ノイズ周波数**

ノイズの平均周波数を、レジスタ6に設定します。設定値の範囲は0～31です。設定するノイズ周波数をFN [Hz]、設定値をR6とすると、R6=111860.78125/FNとなります。ノイズ周波数は、すべてのチャンネルで共通です。

● **チャンネル設定**

音とノイズを出すかどうかをチャンネルごとに設定します。レジスタ7のbit0～2がチャンネル1～3の音に、bit3～5がチャンネル1～3のノイズに対応しており、各bitが0のときに音やノイズが出力されます。bit6～7は固定です。

7	6	5	4	3	2	1	0
1	0	N3	N2	N1	T3	T2	T1

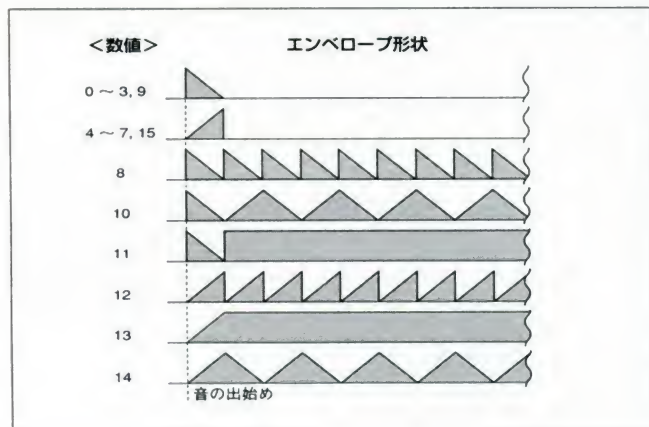
T1: チャンネル1の音
T2: チャンネル2の音
T3: チャンネル3の音
N1: チャンネル1のノイズ
N2: チャンネル2のノイズ
N3: チャンネル3のノイズ

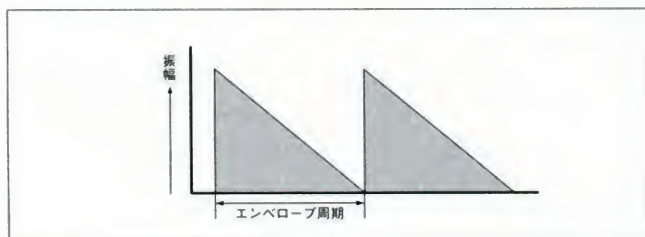
● **音量**

音量をチャンネルごとに設定します。チャンネル1～3がレジスタ8

~10に対応しています。設定値の範囲は0~16ですが、16を指定すると音量はエンベロープ形状とエンベロープ周期に従って変化します。0~15を指定した場合、音量は一定です(15が最大音量で0は無音)。

● エンベロープ形状

エンベロープ形状(音量変化の波形)を、レジスタ13に設定します。設定値の範囲は0~15です。設定値とエンベロープ形状の対応については、下図を参照してください。エンベロープ形状は、すべてのチャンネルで共通です。

● エンベロープ周期

エンベロープ周期(音量変化の周期)を、レジスタ11~12に設定します。設定値の範囲は1~65535で、レジスタ11、12をそれぞれ下位8bit、上位8bitとして設定します。エンベロープ周期は(143 × 設定値) μ秒となります。エンベロープ周期は、すべてのチャンネルで共通です。

参照 PLAY [1]

SPACE$(スペース ダラー)関数

機能 指定した数の空白文字を得る

書式 SPACE$(<数式>)

文例 A$=SPACE$(5)

解説 <数式>で指定する数の空白を持つ文字列を得ます。<数式>に指定できる値の範囲は0~255です。

参照 SPC, STRING$

SPC(スペース)関数

機能 指定した数の空白文字を出力する

書式 SPC(<数式>)

文例 PRINT "ABC";SPC(3);"GHI"

実行結果 ABC GHI

解説 <数式>で指定する数の空白を出力します。<数式>に指定できる値の範囲は0~255です。SPCは、PRINTやPRINT#などの出力文中でのみ使用できます。

参照 SPACE$, TAB

SPRITE$(スプライト ダラー)システム変数

機能 スプライトパターンを定義、参照する

書式 SPRITE$(<スプライトパターン番号>)

文例 SPRITE$(0)=STRING$(32,&HFF)
S$=SPRITE$(I)

解説 スプライトパターンを保持するシステム変数です。スプライトパターンの定義と参照に使用します。スプライトパターンの定義は、SPRITE$への代入によって行います。

<スプライトパターン番号>には、スプライトサイズが8×8ドットの場合は0~255、16×16ドットの場合は0~63の値を持つ数式を指定します(スプライトサイズについてはSCREENを参照)。

スプライトパターンは、8文字(スプライトサイズが8×8ドット)か32文字(16×16ドット)の文字列で表現します。各文字は、スプライトパターン中の横8ドットのうち、ドットのある部分を1、ない部分を0とする値(0~255)をキャラクタコードとする文字となります。スプライトサイズが8×8ドットの場合は、各文字がスプライトパターンの1ラインに対応しており、8ラインで8文字となります。16×16ドットの場合、8×8のパターンを表す文字列を左上、左下、右上、右下の順に並べた文字列になります。

参照 PUT SPRITE, COLOR SPRITE, COLOR SPRITE$, SCREEN

● サンプルプログラム

・8×8ドットのスプライトパターンを定義する

```
10 SCREEN 2,0
20 FOR I=0 TO 7
30 READ A$:SP$=SP$+CHR$(VAL("&B"+A$))
40 NEXT
50 SPRITE$(0)=SP$
60 PUT SPRITE 0,(124,92),15,0
70 I$=INPUT$(1)
80 END
100 DATA 00011000
110 DATA 00111100
120 DATA 01111110
130 DATA 11111111
140 DATA 00011000
150 DATA 00011000
160 DATA 00011000
170 DATA 00011000
```

・16×16ドットのスプライトパターンを定義する

```
10 SCREEN 2,2
20 FOR I=0 TO 15
30 READ A$
31 L$=L$+CHR$(VAL("&B"+LEFT$(A$,8)))
32 R$=R$+CHR$(VAL("&B"+RIGHT$(A$,8)))
40 NEXT
50 SPRITE$(0)=L$+R$
```

```
60 PUT SPRITE 0,(124,92),15,0
70 I$=INPUT$(1)
80 END
100 DATA 0000011111100000
110 DATA 0001100000011000
120 DATA 0010000000000100
130 DATA 0100001111000010
140 DATA 0100010000100010
150 DATA 1000100000010001
160 DATA 1001000110001001
170 DATA 1001001001001001
180 DATA 1001001001001001
190 DATA 1001000110001001
200 DATA 1000100000010001
210 DATA 0100010000100010
220 DATA 0100001111000010
230 DATA 0010000000000100
240 DATA 0001100000011000
250 DATA 0000011111100000
```

SPRITE OFF(スプライトオフ)ステートメント

機能 スプライトの衝突に対応する割り込み処理を禁止する

書式 SPRITE OFF

文例 SPRITE OFF

解説 スプライトが重なったときに実行される割り込み処理の実行を、禁止します。

参照 SPRITE ON, SPRITE STOP, ON SPRITE GOSUB

SPRITE ON(スプライトオン)ステートメント

機能 スプライトの衝突に対応する割り込み処理を許可する

書式 SPRITE ON

文例 SPRITE ON

解説 スプライトが重なったときに実行される割り込み処理の実行を、許可します。

参照 SPRITE OFF, SPRITE STOP, ON SPRITE GOSUB

SPRITE STOP(スプライトストップ)ステートメント

機能 スプライトの衝突に対応する割り込み処理を保留する

書式 SPRITE STOP

文例 SPRITE STOP

解説 スプライトが重なったときに実行される割り込み処理の実行を、保留します。保留された割り込み処理は、SPRITE ONが実行されたときに実行されます。なお、SPRITE STOPは、SPRITE ON状態で実行した場合にのみ有効です。

参照 SPRITE ON, SPRITE OFF, ON SPRITE GOSUB

SQR(スクエアルート)関数

機能 平方根(スクエアルート)を得る

書式 SQR(<数式>)

文例 PRINT SQR(5)

実行結果 2.2360679774998

解説 <数式>の平方根(スクエアルート)を得ます。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数になります。

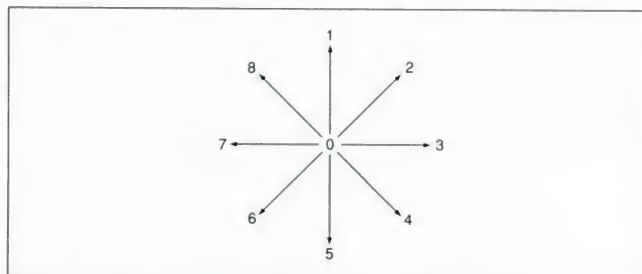
STICK(スティック)関数

機能 カーソルキーやジョイスティックの入力方向を得る

書式 STICK(<ジョイスティック番号>)

文例 S=STICK(0)

解説 <ジョイスティック番号>で指定するジョイスティックの入力されている方向を調べ、その値を得ます。<ジョイスティック番号>には、0～2の値を持つ数式を指定します。0はカーソルキー、1～2はそれぞれポート1、2に接続されたジョイスティックに対応します。

入力方向は0～8の整数で得られます。0は方向が入力されていない状態に対応し、1～8はそれぞれ上、右上、右、右下、下、左下、左、左上に対応します(下図を参照)。

参照 STRIG

● サンプルプログラム

```
10 SCREEN 4,2
20 SPRITE$(0)=STRING$(32,255)
30 PUT SPRITE 0,(X,Y),15,0
40 IF STRIG(0) THEN END
50 S=STICK(0)
60 IF S=1 THEN Y=(Y+191) MOD 192
70 IF S=5 THEN Y=(Y+1) MOD 192
80 IF S=3 THEN X=(X+1) MOD 256
90 IF S=7 THEN X=(X+255) MOD 256
100 GOTO 30
```

STOP(ストップ)ステートメント

機能 プログラムの実行を停止する

書式 STOP

文例 STOP

解説 プログラムの実行を停止し、コマンドレベルに戻ります。END文と異なり、STOP文はファイルを閉じません。停止したプログラムは、CONT文によって実行を再開できます。

参照 CONT, END

STOP OFF(ストップオフ)ステートメント

機能 [CTRL]+[STOP]の押下に対応する割り込み処理を禁止する

書式 STOP OFF

文例 STOP OFF

解説 [CTRL]+[STOP]が押されたときに実行される割り込み処理の実行を、禁止します。

参照 STOP ON, STOP STOP, ON STOP GOSUB

STOP ON(ストップ オン)ステートメント

機能 **[CTRL]+[STOP]の押下に対応する割り込み処理を許可する**
書式 STOP ON
文例 STOP ON
解説 [CTRL]+[STOP]が押されたときに実行される割り込み処理の実行を、許可します。
参照 STOP OFF, STOP STOP, ON STOP GOSUB

STOP STOP(ストップ ストップ)ステートメント

機能 **[CTRL]+[STOP]の押下に対応する割り込み処理を保留する**
書式 STOP STOP
文例 STOP STOP
解説 [CTRL]+[STOP]が押されたときに実行される割り込み処理の実行を、保留します。保留された割り込み処理は、STOP ONが実行されたときに実行されます。なお、STOP STOPは、STOP ON状態で実行した場合にのみ有効です。
参照 STOP ON, STOP OFF, ON STOP GOSUB

STRIG(エス トリガ)関数

機能 **スペースキーやジョイスティックのトリガ状態を得る**
書式 STRIG(<ジョイスティック番号>)
文例 T=STRIG(0)
解説 <ジョイスティック番号>で指定するジョイスティックのトリガの状態を調べ、押されていれば真(−1)を、押されていなければ偽(0)を得ます。<ジョイスティック番号>には、0～4の値を持つ数式を指定します(下表)。

ジョイスティック番号	意 味
0	スペースキー
1	ポート1に接続されたジョイスティックのトリガ1
2	ポート2に接続されたジョイスティックのトリガ1
3	ポート1に接続されたジョイスティックのトリガ2
4	ポート2に接続されたジョイスティックのトリガ2

参照 STICK, ON STRIG GOSUB, STRIG ON, STRIG OFF, STRIG STOP

STRIG OFF(エス トリガ オフ)ステートメント

機能 **スペースキーやジョイスティックのトリガの押下に対応する割り込み処理を禁止する**
書式 STRIG(<ジョイスティック番号>) OFF
文例 STRIG(0) OFF
解説 ジョイスティックのトリガが押されたときに実行される割り込み処理の実行を、禁止します。<ジョイスティック番号>と割り込みを禁止するトリガの対応については、STRIGを参照してください。
参照 STRIG ON, STRIG STOP, ON STRIG GOSUB, STRIG

STRIG ON(エス トリガ オン)ステートメント

機能 **スペースキーやジョイスティックのトリガの押下に対応する割り込み処理を許可する**
書式 STRIG(<ジョイスティック番号>) ON
文例 STRIG(1) ON
解説 ジョイスティックのトリガが押されたときに実行される割り込み処理の実行を、許可します。<ジョイスティック番号>と割り込みを許可するトリガの対応については、STRIGを参照してください。
参照 STRIG OFF, STRIG STOP, ON STRIG GOSUB, STRIG

STRIG STOP(エス トリガ ストップ)ステートメント

機能 **スペースキーやジョイスティックのトリガの押下に対応する割り込み処理を保留する**
書式 STRIG(<ジョイスティック番号>) STOP
文例 STRIG(2) STOP
解説 ジョイスティックのトリガが押されたときに実行される割り込み処理の実行を、保留します。<ジョイスティック番号>と割り込みを保留するトリガの対応については、STRIGを参照してください。保留された割り込み処理は、STRIG ONが実行されたときに実行されます。なお、STRIG STOPは、STRIG ON状態で実行した場合にのみ有効です。
参照 STRIG ON, STRIG OFF, ON STRIG GOSUB, STRIG

STR$(ストリング ダラー)関数

機能 **数値を表す文字列を得る**
書式 STR$(<数式>)
文例 PRINT STR$(−3.14)
実行結果 −3.14
解説 <数式>の値を10進表記の文字列に変換し、その結果を得ます。得られる値は文字型です。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定できます。
参照 VAL, STRING$, BIN$, OCT$, HEX$

STRING$(ストリング ダラー)関数

機能 **指定の文字を指定の数だけ得る**
書式 STRING$(<文字数>, <文字式>)
STRING$(<文字数>, <数式>)
文例 PRINT STRING$(5,"A");STRING$(3,&H40)
実行結果 AAAAA@@@
解説 <文字式>または<数式>で指定する文字を、<文字数>の数だけ連ねた文字列を得ます。得られる値は文字型です。

<文字式>を指定した場合、その先頭文字のみが有効です。グラフィックキャラクタを指定すると、CHR$(1)を<文字数>の数だけ得ます。

<数式>を指定した場合、その値(0～255)をキャラクタコードとみなし、対応する文字を<文字数>の数だけ得ます。
参照 SPACE$, STR$

SWAP(スワップ)ステートメント

機能 **変数の値を交換する**
書式 SWAP <変数名1>, <変数名2>
文例 SWAP A,B
解説 <変数名1>と<変数名2>の値を交換します。<変数名1>、<変数名2>に指定する変数には、整数型、単精度実数型、倍精度実数

型、文字型のいずれも使用できますが、両者の型が完全に一致していなければなりません。

TAB(タブ)関数

機能 **指定の桁まで空白を出力する**

書式 TAB(<数式>)

文例 PRINT "ABC";TAB(6);"GHI"
実行結果　ABC　GHI

解説 カーソル位置から<数式>で指定する桁まで、空白を出力します。<数式>に指定できる値の範囲は0～255ですが、指定の桁数が出力先の桁数を越える場合、折り返して出力します。現在の出力位置の桁数が<数式>に指定した値よりも大きいと、何も出力しません。TABは、PRINTやPRINT#などの出力文中でのみ使用できます。

参照 SPC

TAN(タンジェント)関数

機能 **正接(タンジェント)を得る**

書式 TAN(<数式>)

文例 PRINT TAN(3.14159266/3)
実行結果　1.7320508161158

解説 <数式>の正接(タンジェント)を得ます。<数式>はラジアン単位で、整数型、単精度実数型、倍精度実数型の値を指定できますが、得られる値は倍精度実数です。

参照 ATN, SIN, COS

TIME(タイム)システム変数

機能 **1/60秒ごとに増加するカウンタを参照、変更する**

書式 TIME

文例 T=TIME

解説 1/60秒ごとに1ずつ増加するシステム変数です。TIMEは0～65535の値を持ち、参照と代入が可能です。0から65535まで増加した後は、再び0に戻り、増加を続けます。

TIMEはVDPの垂直帰線割り込みによって更新されるため、フロッピーディスクドライブのアクセス中などの割り込みが禁止されている期間は増加しません。

参照 GET TIME, SET TIME, SET DATE, GET DATE

TRON(トレースオン)ステートメント

機能 **プログラムの状態追跡機能を有効にする**

書式 TRON

文例 TRON

解説 BASICが持っている、プログラムの実行状態を追跡する機能を有効にします。TRONを実行すると、以後、実行したプログラムの行番号が角括弧に囲まれて表示されます。

状態追跡機能を無効にするには、TROFFかNEWを実行します。

参照 TROFF, LIST, RUN

TROFF(トレースオフ)ステートメント

機能 **プログラムの状態追跡機能を無効にする**

書式 TROFF

文例 TROFF

解説 プログラムの実行状態を追跡する機能を無効にします。

参照 TRON, LIST, RUN

VAL(バリュー)関数

機能 **文字列の表す数値を得る**

書式 VAL(<文字式>)

文例 PRINT VAL("-3.14");VAL(" &HFF")
実行結果　-3.14　255

解説 <文字式>で指定する文字列を数値に変換し、その結果を得ます。<文字式>から空白を取り除いた文字列が、+、-、&、数字(0～9)以外の文字で始まっている場合、常に0を得ます。

参照 STR$, BIN$, OCT$, HEX$

VPEEK(ブイピーク)関数

機能 **VRAM上の指定された番地の値を得る**

書式 VPEEK(<数式>)

文例 A=VPEEK(0)

解説 <数式>で指定する番地の値をVRAMから読み出し、その結果を得ます。得られる値は0～255の整数です。番地はSET PAGEで指定するアクティブページの先頭からのオフセットです。<数式>には、整数型、単精度実数型、倍精度実数型の値を指定でき、その値の範囲は-32768～32767(または0～65535)となります。

参照 VPOKE, SET PAGE, PEEK, POKE

VPOKE(ブイポーク)ステートメント

機能 **VRAM上の指定された番地にデータを書き込む**

書式 VPOKE <番地>,<データ>

文例 VPOKE &HF3DB,0

解説 <番地>で指定するVRAM上の1バイトに<データ>の値を書き込みます。<番地>はSET PAGEで指定するアクティブページの先頭からのオフセットです。<番地>には、整数型、単精度実数型、倍精度実数型の値を指定でき、その値の範囲は-32768～32767(または0～65535)となります。<データ>に指定できる値の範囲は0～255の整数です。

参照 VPEEK, SET PAGE, POKE, PEEK

WIDTH(ウィドス)ステートメント

機能 **画面の桁数を設定する**

書式 WIDTH <桁数>

文例 WIDTH 32

解説 画面の桁数を設定します。テキストモード(SCREEN 0～1)でのみ、有効です。

<桁数>に指定できる値の範囲は、SCREEN 0で1～80(MSX1では1～40)、SCREEN 1では1～32です。初期値に、SCREEN 0で39、SCREEN 1で29が設定されています。

参照 SCREEN

エラーコード表

メッセージ	環境	エラーコード	意味
Bad FAT	Disk	60	ディスクの管理領域(FAT：File Allocation Table)に異常がある。
Bad drive name	Disk	62	ドライブ名の指定に誤りがある。ドライブ名はA:～H:を使用する。
Bad file mode	Disk	61	OPEN文で指定したモードと、入出力のモードが異なる。
Bad file name		56	ファイル名が適当でない。
Bad file number		52	指定したファイル番号が、MAX FILESで指定した値を越えている。
Bad sector number	Disk	63	セクタ番号が適当でない。
Can't continue		17	プログラムの実行を再開できない。エラー終了した場合や、プログラムの書き換え、CLEAR文の実行を行った場合は、CONT文で実行を再開することはできない。
Device I/O error		19	デバイス(画面やプリンタなど)への入出力でエラーが発生した。
Direct statement in file		57	LOAD文やRUN文で読み込もうとしたファイルに、プログラム以外のデータが含まれている。
Disk full	Disk	66	ディスク上の空き領域がない。
Disk I/O error	Disk	69	ディスクへの入出力でエラーが発生した。
Disk offline	Disk	70	ディスクがディスクドライブに装着されていない。
Disk write protected	Disk	68	ディスクが書き込み保護されている。
Division by zero		11	0による除算か、0に対する負のベキ乗を行った。
Field overflow	Disk	50	FIELD文によるフィールドサイズの合計が、OPEN文で指定したレコード長を越えた。
File already exists	Disk	65	NAME文で指定した新しいファイル名が、既存のファイル名と重複している。
File already open		54	OPEN文で指定したファイル番号は、既に割り当て済みである。
File not found		53	指定のファイルが存在しない。
File not OPEN		59	指定したファイル番号に、ファイルが割り当てられていない。
File still open	Disk	64	開かれているファイルのファイル名を削除／変更しようとした。
Illegal direct		12	ダイレクトモードでは実行できないステートメントをダイレクトモードで実行しようとした。
Illegal function call		5	ステートメントや関数の使い方(引数の範囲など)が間違っている。
Input past end		55	ファイルの内容をすべて読み込んだ後、更に読み込みを実行した。EOFを参照。
Internal error		51	BASIC内部でエラーが発生した。
Line buffer overflow		25	入力データの大きさが、バッファの大きさを越えている。
Missing operand		24	必須のパラメータが指定されていない。
NEXT without FOR		1	NEXT文に対応するFOR文が存在しない。
No RESUME		21	エラー処理ルーチンにRESUMEが存在しない。
Out of DATA		4	READ文で読み出すデータがない。
Out of memory		7	メモリが足りない。プログラムが長すぎたり、配列変数が大きすぎたりすると発生する。
Out of string space		14	文字列を格納する領域が足りない。
Overflow		6	演算結果が、型に許される範囲を超えた。
Redimensioned array		10	配列変数が二重に定義された。
Rename across disk	Disk	71	異なるディスクドライブ間で、NAME文を実行した。
RESUME without error		33	エラーが発生していないにも関わらず、RESUMEが実行された。
RETURN without GOSUB		3	RETURNに対応するGOSUBが存在しない。
Sequential I/O only		58	シーケンシャルファイルに対してランダムアクセスを行おうとした。
String formula too complex		16	指定した文字列が複雑すぎるため、文字列を分割する必要がある。
String too long		15	文字列の長さが255文字を越えた。
Subscript out of range		9	配列変数の添え字が、DIM文で定義した添え字の最大値より大きい。

メッセージ	環境	エラーコード	意味
Syntax error		2	括弧が対応していない、ステートメントや関数のスペルが間違っている、区切り文字がないなど、文法を誤っている。
Too many files	Disk	67	ディレクトリエントリに登録できるファイルの数を越えて、ファイルを作ろうとした。
Type mismatch		13	データの型が一致しない。
Undefined line number		8	指定した行番号がプログラム中に存在しない。
Undefined user function		18	DEF FN文で定義していないユーザー定義関数を使用した。
Unprintable error		23, 26～49, 76～255	メッセージが定義されていないエラー。
Verify error		20	カセットテープに保存されたプログラムと、メモリ上のプログラムが一致しない。

※ DiskはDisk BASICのみで発生することを意味する。

キャラクタコード表

上位4ビット →

下位4ビット ↓

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	↑コントロールキャラクタ↓			0	@	P	‘	p	♠			ー	タ	ミ	た	み
1			!	1	A	Q	a	q	♥	あ	。	ア	チ	ム	ち	む
2			”	2	B	R	b	r	♣	い	「	イ	ツ	メ	つ	め
3			#	3	C	S	c	s	◆	う	」	ウ	テ	モ	て	も
4			$	4	D	T	d	t	○	え	、	エ	ト	ヤ	と	や
5			%	5	E	U	e	u	●	お	・	オ	ナ	ユ	な	ゆ
6			&	6	F	V	f	v	を	か	ヲ	カ	ニ	ヨ	に	よ
7			’	7	G	W	g	w	ぁ	き	ァ	キ	ヌ	ラ	ぬ	ら
8			(	8	H	X	h	x	ぃ	く	ィ	ク	ネ	リ	ね	り
9			)	9	I	Y	i	y	ぅ	け	ゥ	ケ	ノ	ル	の	る
A			*	:	J	Z	j	z	ぇ	こ	ェ	コ	ハ	レ	は	れ
B			+	;	K	[	k	{	ぉ	さ	ォ	サ	ヒ	ロ	ひ	ろ
C			,	<	L	¥	|	¦	ゃ	し	ャ	シ	フ	ワ	ふ	わ
D			−	=	M	]	m	}	ゅ	す	ュ	ス	ヘ	ン	へ	ん
E			.	>	N	^	n	~	ょ	せ	ョ	セ	ホ	゛	ほ	
F			/	?	O	_	o		っ	そ	ッ	ソ	マ	゜	ま	█

↑カーソル

グラフィックキャラクタ

	4	5
0		π
1	月	
2	火	
3	水	
4	木	
5	金	
6	土	
7	日	
8	年	
9	円	
A	時	
B	分	
C	秒	
D	百	大
E	千	中
F	万	小

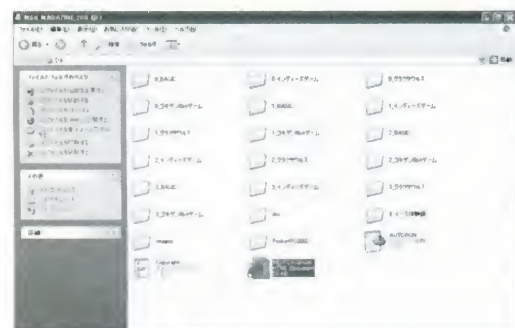
◆グラフィックキャラクタの表示方法

グラフィックヘッダ(&H01)を先頭に付けた計2バイトで表す

例:「日」を表示する場合　PRINT CHR$(&H01);CHR$(&H47)

付属CD-ROMの使い方

CD-ROMドライブに付属CD-ROMを挿入すると、自動的にブラウザが立ち上がり、「付属CD-ROMインストールガイド」が表示される。ブラウザが自動起動しない場合は、エクスプローラでCD-ROMを開き、「MSXPLAYerinst.htm」ファイルをダブルクリックして起動してほしい。

←付属CD-ROMを挿入すると、自動的にブラウザが起動し、「付属CD-ROMインストールガイド」が表示される。

←ブラウザが起動しない場合は、CD-ROM内の「MSXPLAYerinst.htm」ファイルをダブルクリックする。

付属CD-ROMのインストール、および使い方は、本書「スタートMSXPLAYer」、「MSXPLAYer Tips集」に詳しく解説しているので、そちらを参照のこと。なお最新情報は、MSX MAGAZINE公式サイト(http://www.ascii.co.jp/pb/ant/msx/)に掲載されている。

←最新情報が常時、入手できるMSX MAGAZINE公式サイト

また、付属CD-ROMには、作者のご厚意により書籍、雑誌、およびゲームの付属マニュアルの全文、もしくは抜粋をPDFファイルとして収録した。PDFファイルの表示や印刷には、「Adobe Acrobat」もしくは「Adobe Acrobat Reader」(無償)が必要だ。

Acrobat Readerのダウンロードは、Adobe社のサイト(http://www.adobe.co.jp/products/acrobat/readstep.html)より行ってほしい。

←PDFの表示画面

各PDFファイルの原書の一覧を以下に示す。

●castle_man.pdf(全文)
「Catsle Excellent　スーパーヒントブック」
1986年12月8日初版発行
著者：新保剛平・竹山正寿
イラスト：牧野伸康(表紙・本文)
ISBN：4-87148-035-6
発行：株式会社アスキー

●MSXgwb.pdf(抜粋)
「MSX グラフィック・ワークブック」
1984年5月5日初版発行
著者：桜田幸嗣・蓑島聡
イラスト東恩納裕一(本文)
ISBN：4-87148-730-X
発行：株式会社アスキー

●MSX-DOSsh.pdf(抜粋)
「MSX-DOS スーパーハンドブック」
1987年11月9日初版発行
著者：BITS
イラスト：ケロヨン村田(表紙)／茶目高橋(本文)
ISBN：4-87148-045-3
発行：株式会社アスキー

●graphsaurus_man.pdf(全文)
「GRAPHSAURUS ver2.0」マニュアル
1991年発行
発売：株式会社ビッツー

●MSX1992summer.pdf(抜粋)
「MSX MAGAZINE '92夏号」
1992年8月30日発行
著者：北神陽太(特集)
イラスト：岩村実樹(表紙)
発行：株式会社アスキー

プロジェクトEGG ゲームカタログforMSX

MSXPLAYerもソフトがなければ魅力半減。MSX復活にはゲームソフトの復刻が不可欠だが、その先鋒とも言える『プロジェクトEGG』のMSXゲーム群をここで紹介しよう。

EGGで購入できるMSXゲームのタイトルの一覧です。詳しくはホームページをご覧ください。
http://www.soft-city.com/egg/ (2002年11月1日現在)

ARCUS(アークス) for MSX2

日本テレネット
ジャンル：ロールプレイング ¥700

RPGにつきものの経験値やレベルといった概念をなくし、ストーリーを徹底重視にしたドラマチックRPG。ウルフチームの作品。

ARCUSⅡ(アークスⅡ)－Silent Symphony－

日本テレネット
ジャンル：ロールプレイング ¥700

前作から10年後。成長したピクトを主人公にしたドラマチックRPGの続編。ビュー・ポイント・アタック、ネーム・フィードバック・システムといった新システムを多数採用し、前作をより進化させた傑作ソフト。

あかんベードラゴン

ウインキーソフト
ジャンル：シミュレーション ¥600

可愛い怪獣等のキャラクターによるシミュレーション&アクション。戦略なしにはクリアはあり得ないぞ。今も続く人気ゲームの基礎になった作品。

アメリカントラック

日本テレネット
ジャンル：アクション ¥600

爆走するトラック運転手。潰せ！蹴散らせ！ぶっ飛ばせ！でもそれはゲームの世界でだけにしておきましょう。日本テレネットのMSX初参入作品。

EGGY MSX

ボーステック
ジャンル：アクション ¥500

空中飛行可能なボディーアーマーが敵の攻撃をかわしつつパラシュートの物資を回収する。その浮遊感覚と独特のセンスは絶品。ご存知プロジェクトEGGのマスコット。

XZR ～破戒の偶像～

日本テレネット
ジャンル：アクションRPG ¥600

中世の中東を舞台に繰り広げられる宗教的な戦い。オリジナリティを強く感じさせる設定。謎解き・ストーリー・キャラが独特な意欲作。

XZR Ⅱ ～完結編～

日本テレネット
ジャンル：アクションRPG ¥700

何が正義で何が悪なのか。各宗教の異端者が織り成す正義の戦い。現代の悪をいかに倒すか。異色作XZRシリーズ完結編。

EMMY Ⅱ

工画堂スタジオ／アスキー
ジャンル：その他 ¥700

MSX初？の人工知能のゲーム。可愛いエミーと仲良くなろう。エミーはMSXのことも知ってるみたいだよ。さあ二人きりで会話を楽しみましょう。会話を楽しむうちに徐々にスムーズな会話になるはず。

MSXTraiN ～MSXトレイン～

モオソフト／ファミリーソフト
ジャンル：その他 ¥700

モオソフトが贈る「RM」、「紅のイジャン」など5つのゲームが収録されたディスクマガジン形式の作品。これまた豪華なキャラクターデザイン陣。

合体ヴォルカード

dB-SOFT
ジャンル：シューティング ¥700

エネルギーがFULLになると僚機が合体してヴォルガードになる。合体シーン見たさにじっと耐えろ！衝撃の超メカ合体シューティングゲーム。

Xak ～The Art of Visual Stage～

マイクロキャビン
ジャンル：アクションRPG ¥700

「VRシステム」と呼ばれる視覚的表現を実現することで先進のRPGを誕生させたマイクロキャビンの意欲作。続編、外伝を含み、全5作品がリリースされた人気シリーズの第一弾。

神の聖都

パンサーソフト
ジャンル：アドベンチャー ¥800

膨大なグラフィック数、本格派BGM。妖艶なる謎の美女も登場するハイパービジュアルアドベンチャー。付録CDが一緒にダウンロードできます。

The Tower? of CABIN

マイクロキャビン
ジャンル：アドベンチャー ¥500

マイクロキャビンを舞台として展開されるアドベンチャーゲーム。当時のスタッフのメッセージやミニゲームの数々、実験的作品やXakのサイドストーリーが楽しめるお祭り的内容。キャビン・ファン必須の1本。

シュヴァルツシルト ～狂嵐の銀河～

工画堂スタジオ
ジャンル：シミュレーション ¥700

宇宙の彼方で銀河を股に掛けた星間戦争が！1989年に発売された星間戦争をモチーフにしたSLG。他機種も合わせ14作品もリリースされた大人気ロングセラー本格シミュレーションの原点。

白と黒の伝説 ～百鬼編～

スタジオWING
ジャンル：アドベンチャー ¥600

白と黒それは神々の果てしない戦いの物語。その一部をゲーム化。ゲームと連動する音楽等アイデア満載。壮大な構想、人気シリーズの第一部。『サイフォニック・アドベンチャーゲーム』が17年ぶりに復刻。

ゼクサス 光速2000光年

dB-SOFT
ジャンル：シューティング ¥600

初期MSXで3Dシューティングゲームというジャンルに挑戦した意欲作。ワイヤーフレームで表現された戦闘シーンの迫力は一見の価値あり。

ZEXAS limited

dB-SOFT
ジャンル：シューティング
¥700

好評を博したゼクサス光速2000光年の続編。テープからROMに変わり立ち上げが楽に。グラフィックも大幅にグレードアップ。初期MSXの限界に挑戦したシューティングゲーム。

反生命戦機アンドロギュヌス

日本テレネット
ジャンル：シューティング
¥700

世にも珍しい下方向へ進む縦スクロールシューティングで、ひたすら下にスクロールする。タイトルのアンドロギュヌスとは両性具有という意味で、ゲームの途中で男性キャラから女性キャラに変化し、戦略にも変化が。

龍の花園

モオソフト／ファミリーソフト
ジャンル：アドベンチャー
¥700

場所、時間、行動。主人公の行動によってエンディングが変わるマルチエンディング方式を採用した超難解ADV。君はすべてのストーリーを見ることができるか？豪華原画陣。

Niko2 ～ニコニコ～

日本テレネット
ジャンル：アクション
¥600

愛らしいキャラクターが微笑ましい対戦型2Dアクションゲーム。ボールをたくさん自分のゴールに入れた方が勝ち。リアルタイムで起こるキャラの拡大縮小や6種類の効果を発揮する人参がゲームを盛り上げる。

フレイMSX2

マイクロキャビン
ジャンル：アクションRPG
¥600

あのXakのヒロイン、フレイが主役。カワイイARPGになりました。大好きなラトクを追いかけて冒険の旅。果たして無事に逢えるのか。愛と感動と笑いに満ちた、コミカルタッチXak外伝ストーリー。

レイドック

T＆Eソフト
ジャンル：シューティング
¥600

美麗なグラフィック＆スクロール、1ドット単位の当り判定。当時のゲーマーを唸らせた作品。コレをやるためにMSX2買った人、手を上げてー。

レイドック2 ～LAST ATTACK～

T＆Eソフト
ジャンル：シューティング
¥700

縦、横、斜めの滑らかなスクロール、自然画モードの画面。シリーズの完結編。MSX2+専用のキラータイトルとして登場した。コレをやるためにMSX2+買った人、手を上げてー。

ハイドライド

T&Eソフト
ジャンル：アクションRPG
¥500

勇者は謎を解き明かしフェアリーランドに平和を取り戻すことができるのか。その後の多くのRPGに大きな影響を与えた大人気シリーズ第一作。

夢幻戦士ヴァリス

日本テレネット
ジャンル：アクション
¥600

美少女キャラを主人公にしたアクションゲーム。1ステージ最大1024画面の広大なマップが待ち受ける。

ハイドライド MSX2

T&Eソフト
ジャンル：アクションRPG
¥600

1984年に発売された"アクションRPG"の元祖。MSX版より美しい画面が売り。ちょっとめずらしいVRAM 64KBモードを搭載。もちろんVRAM 128KBフルスペックのMSX2モードにも対応。

夢幻戦士ヴァリスII

日本テレネット
ジャンル：アクション
¥700

MSX2初の本格多重スクロールと美しいビジュアル画面。MSXにたくさんの作品を提供した日本テレネットの代表作。アイテム画面でヒロイン優子を着せ替えできるシステムが好評。

レリクス

ボーステック
ジャンル：アクション
¥0

目的、ストーリー、キャラクターなどすべてが謎。与えられたのは操作方法だけ。多関節キャラがグリグリ動くサイバーアクションゲーム。

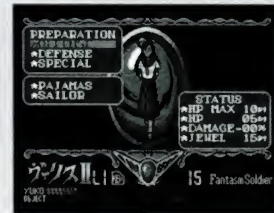
レリクス MSX2

ボーステック
ジャンル：アクション
¥500

漢字表示のストーリーがオープニングとエンディングに付属。グラフィックが綺麗になり動きもスムーズになって再登場。やっとこれでストーリーが判った？

妖怪探偵ちまちま

ボーステック
ジャンル：アクション
¥500

コミカルな主人公の動きと、火の玉の微妙な操縦が人気の秘密。プログラムコンテスト最優秀賞受賞作品がMSXに。

らぷてっく2

dB-SOFT
ジャンル：パズル
¥800

独特な軌道を描くジャンプを駆使して所定のブロックの上を通るだけの単純なルールのパズルゲーム。けど奥が深いのよコレが。

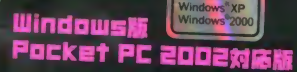
10年ぶりに復活したMSXマガジンなんだけど、どうだったかな？MSXとMSXPLAYerの歴史やこれからのこと、名作ゲーム徹底紹介、さらにはBASIC解説など盛りだくさんの内容＋付録のCD-ROM＋特製シールで、蘇ったMSXのパワーを感じてもらえたことと思う。というわけで、編集部ではみなさんの応援を待っている。ご意見、ご希望、自作プログラムなどをどんどん送ってきてほしいのだ。

発行人	小森 哲郎
編集人	土屋 信明
編集長	佐藤 英一
編集	竹内 充彦
	秋山 耕一
	二瓶 朗
	白須 勝海
	加瀬 典子
	遠藤 さちえ
	林 亜紀子
表紙デザイン	川之上 しのぶ
制作	アスキー書籍制作部
	野村 亮
	川之上 しのぶ
レイアウトデザイン	壁谷沢 敦子
	木村 明咲子
	加納 香代子
	白須 勝海
撮影	石原 秀樹（石原写真事務所）
進行管理	辻 憲二
出版営業	松本 浩
	池田 一樹
	鹿野 恵子
宣伝グループ	山田 浩之
	山嵜 夏子
生産管理部	平井 康則
印刷	大日本印刷株式会社

●落丁、乱丁本は、送料負担にてお取り替えいたします。お手数ですが、小社営業部までご返送ください。

●Web読者アンケートのお知らせ
読者の皆さまのご意見を今後の企画・編集の貴重な資料とさせていただきますので、Web読者アンケートにご協力ください。

http://mkt.uz.ascii.co.jp/

回答者の方に抽選でプレゼントを提供しています。詳しくはWebをご覧ください。

MSX MAGAZINE公式ホームページ：
http://www.ascii.co.jp/pb/ant/msx/

メールアドレス：
ant-msx@ml.ascii.co.jp
※ご返事ができない場合、もしくはお時間がかかる場合がありますので、あらかじめご了承ください。

MSXアソシエーション
浦田 真司、柏崎 太郎、日下 義政、後藤 啓太、関根 義仁、ときしろいずみ、中山 淳、新留 知洋、羽賀 義典、花岡 朋和、宮下 勝、横居 英克

Special Thanks：
PASTEL HOPE、Teamぷらんくとん、αH7、GIGAMIX、うえのまさひろ、まかろん、JL2TBB、yambal、MAB、K.N.、横川 理彦（よっちゃん）、池田 国治、坂上 彰、阿部 聡、gyro、ゆっこ、MZK、小林 崇、Bernard Lamers、and MSXマガジン復刊準備号1,2スタッフ、imahi、白神裕久、MSXユーザーグループ

PocketPC機材協力：
WindowsCE FAN (http://www.wince.ne.jp/)

2002年12月31日　初版発行

編　者　アスキー書籍編集部
発行所　株式会社アスキー
〒160-8584　東京都新宿区信濃町34番地
書籍関連サイト　http://www.ascii.co.jp/books/
出版営業部　(03)5362-3300（ダイヤルイン）
ANT書籍編集部　FAX：(03)5362-3398

ISBN4-7561-4210-9　Printed in Japan
●1322748

■付属CD-ROMの利用環境

本CD-ROMのインストール方法、および使い方は、本書「スタートMSXPLAYer」をご覧ください。動作環境は、以下のとおりです。

●Windows版MSXPLAYer推奨スペック

OS	Windows98／98SE／Me／2000／XP（2000／XP推奨）
CPU	Celeron500MHz以上（800MHz以上推奨）
サウンドカード	DirectX5以降対応のサウンドカード
ビデオカード	16ビットカラー、800×600ドット以上の表示が可能なもの

●PocketPC2002対応版MSXPLAYer

PocketPC2002仕様の各機種（現行の最新機種を推奨）
※注意：ただし、機種により制限事項があります。本書60ページを必ずお読みの上、インストールと実行を行ってください。

■ご利用にあたっての注意事項

本製品は、株式会社アスキーが著作権者の許諾を受け、お客様に使用許諾するものです。本製品は、日本国内においてのみ使用できます。
本製品のご利用にあたっては、下記の注意事項をお読みください。

①お客様は、このCD-ROM内のプログラム、およびデータをお客様が保有されるコンピュータ機器上にて、１台に限って稼働させ、ご利用いただけます。複数台にて同時にご利用いただくことはできません。
②CD-ROM内のデータは、お客様が個人利用の範囲内において使用できます。
③お客様は、CD-ROM内のデータの全部、または一部を第三者に譲渡、貸与、自動公衆送信（送信可能化を含む）することはできません。
④お客様がCD-ROM内のデータを運用した結果生じた損害については、株式会社アスキーおよび著作権者は、一切責任を負いかねますのでご了承ください。

■ Windows対応版ソフト

ゴキゲン8bitゲーム
(P.10からを参照)

- ●ポコスカウォーズ
- ●ザ・キャッスル
- ●キャッスルエクセレント
- ●テセウス
- ●ベアーズ
- ●ローターズ
- ●イリーガス
- ●スコープオン
- ●たわらくん
- ●トライアルスキー
- ●ハイパニック
- ●ライズアウト
- ●レッドゾーン
- ●テグザー
- ●ファイアーホーク
- ●クインプル
- ●ファミクルパロディック2

インディーズゲーム
(P.48からを参照)

- ●マジカルラビリンス Remix
- ●はいっぷる
- ●無敵戦士ヤジウマン
- ●ASTRO MONSTERS

BASIC
(P.152を参照)

- ●マウス対応BASIC
- ●ジョイパッド対応BASIC

イースシリーズ体験版（EGG版）
(P.46を参照)

- ●イース
- ●イース II
- ●ワンダラーズ フロム イース

それぞれ20分間プレイ可能な体験版です。製品版は、EGGサイトで購入できます。

グラフサウルス Ver 2.0
(P.102を参照)

- ●グラフサウルス Ver 2.0

■ Pocket PC 2002対応版収録ソフト

収録ソフト
(P.10からを参照)

- ●ポコスカウォーズ
- ●たわらくん
- ●はいっぷる
- ●ベアーズ
- ●トライアルスキー
- ●ローターズ
- ●ハイパニック
- ●イリーガス
- ●クインプル

ISBN4-7561-4210-9

C3004 ¥2500E

定価 本体2,500円 +税

9784756142108

1923004025000